DAWSON COLLEGE – Electronics Technology Department WINTER 2012

Computer Network Project (243-698-DW)

Internet Thermostat

Francis Bergin

May 14, 2012

1 Contents

2 Table of figures

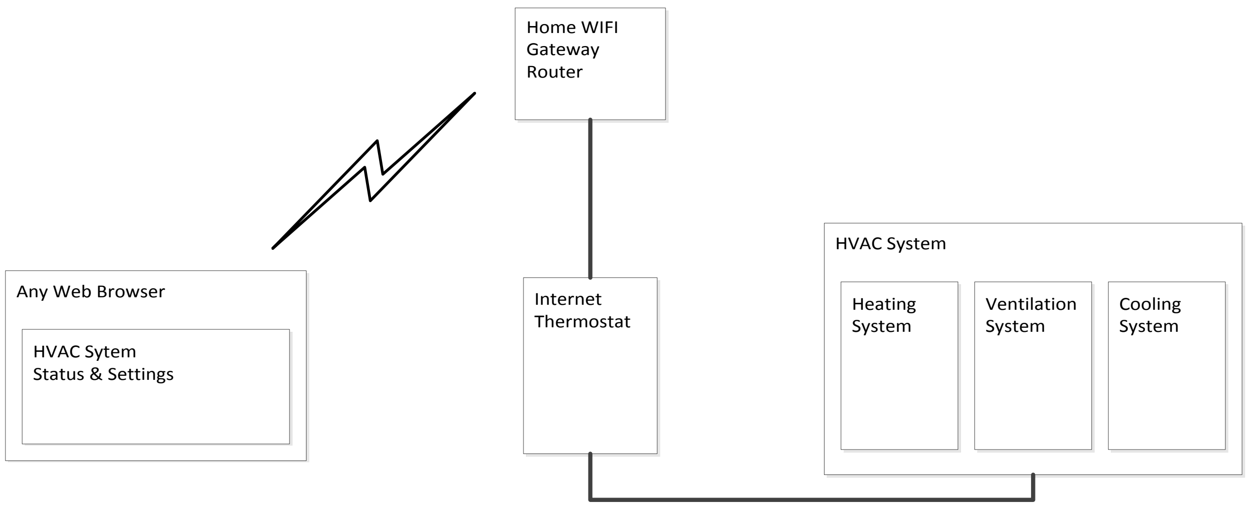
3 Introduction

3.1 Description

The Internet Thermostat is a feature-rich thermostat that can be used to accurately measure temperature of surrounding areas and control a complete HVAC system including heating, ventilation, and air conditioning. Its internet connection allows easy monitoring and adjustments of the whole system from any local or remote location using a web browser.

3.2 Block Diagram

Figure 4.1.1-1: Internet Thermostat Block Diagram



3.3 Specifications

- Input

  - Input voltage: 6-30VAC/DC

- Temperature

  - Operating temperature range: -30°C to 70°C

- Network

  - Network protocols: TCP, IP, ARP, ICMP, HTTP

  - 10Base-T

  - Auto MDIX

- Supported Browsers

  - Safari, Chrome, Firefox, Opera

- Output

  - Relay switching capacity: 200V, 0.5A, 10W

4 User Instructions

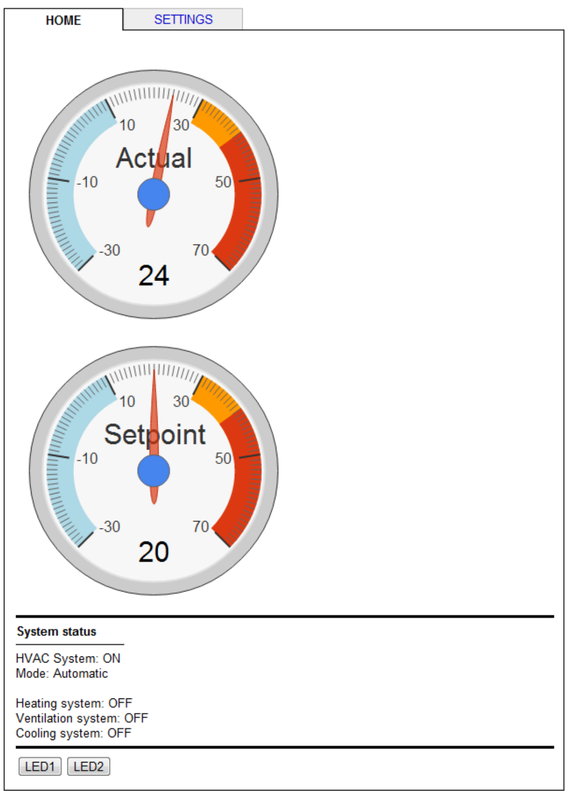
4.1 User Settings

4.1.1 Changing the set point temperature using the dial

The dial to the right of the display is used to increase and decrease the set point temperature by turning it clockwise and counterclockwise respectively. This dial can cause errors in the setting of the temperature when it is rotated too quickly. When turning this dial, you will be able to view the changes made to the set point temperature.

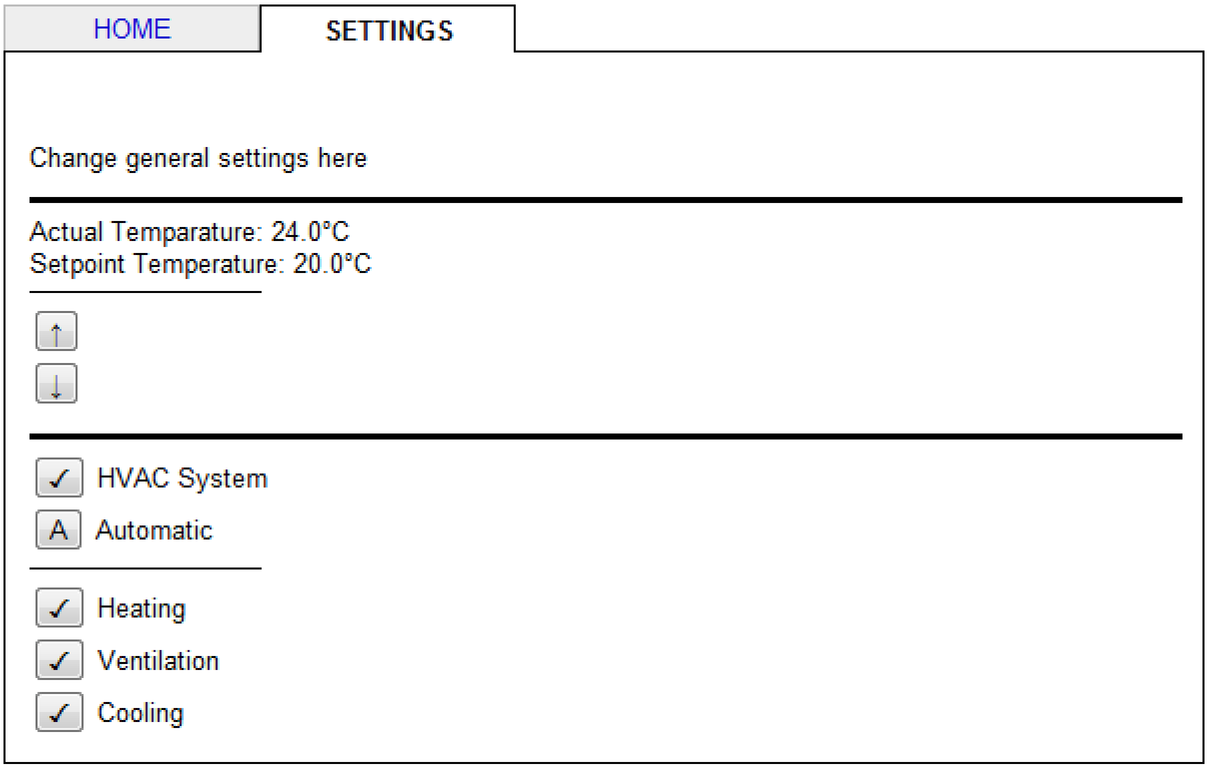
4.1.2 Navigating the device web page

Figure 4.1.2-1: Web Interface - Home



On the home page of the Internet Thermostat web page, there is a view of the temperature settings as well as a System status section. The first gauge presented represents the actual temperature in degrees Celsius. This is the temperature calculated by the temperature sensor. The second gauge displays the set point or desired temperature which can be set by the user. The default set point value is 20 degrees. The system status sections shows which parts of the device are turned on and off. It also shows if the HVAC system is set to manual mode or automatic mode.

Figure 4.1.2-2: Web Interface - Settings



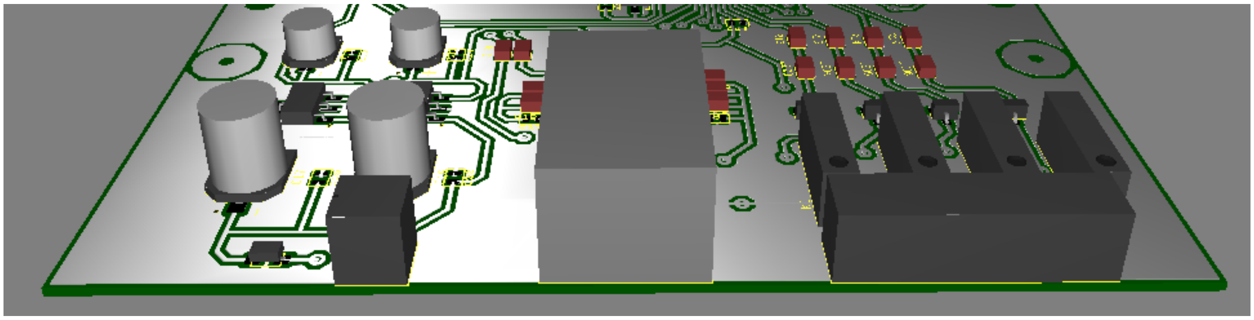
On the setting screen, you can increase and decrease the set point temperature. The HVAC system button will shut down completely all heating, cooling and ventilation. When the device is in automatic mode, it will control heating, cooling, and ventilation automatically, based on actual temperature measurements and set temperature. When in manual mode, the user can decide which part is on or off. This is mode is not recommended for heating and cooling as it could make the system work for a very long time if forgotten. If prolonged ventilation is needed in the areas, this mode would be good.

The heating, ventilation, and cooling can also be turned on and off independently in automatic mode. This would be good for turning off the heating system during summer so that at night if the actual temperature goes a little below the set point temperature, the heating system does not start.

5 Installation and Calibration

5.1 Connections

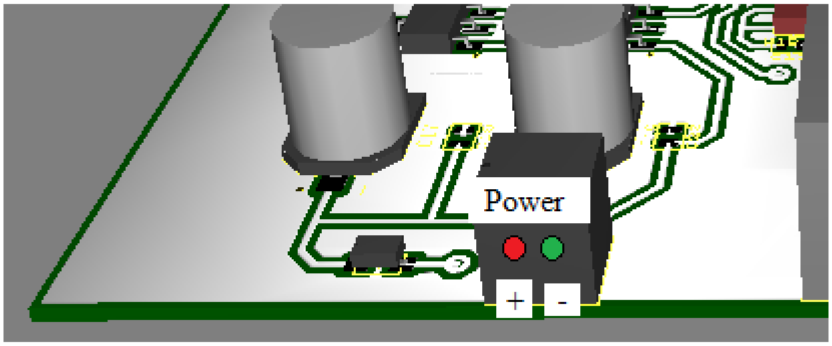
Figure 4.1.2-1: Board Connections



5.1.1 Connecting to power

To get started with the internet thermostat, first connect the device to a 12 or 24V, AC or DC power supply making sure the ground side is to the right as shown in Figure 5.1.1-1.

Figure 5.1.1-1: Power Connections

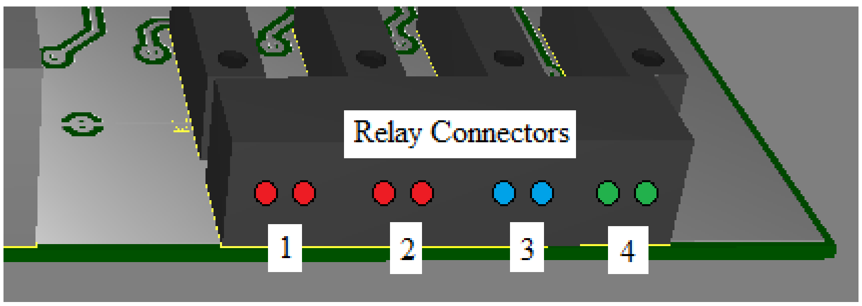


Tighten the connector screws and make sure the wires are held firmly. Once this is done, you can plug in the power supply. The device turns on right away; the display as well as the relay lights will be light up for about one second until the device finishes calculating an average ambient temperature. It is normal that the first few temperature readings are lower than expected; this is to compensate for gradual board heat.

5.1.2 Connecting to HVAC system

On the Internet Thermostat, there are four relay interfaces capable of switching on and off a load of up to 10W. These interfaces can be used to connect to bigger relays which will in turn control heating, cooling and ventilation. As shown in Figure 5.1.2-1, there are two relays used for heating: the first one is for backup heating and the second is used for the primary heater. The third connector is for the air conditioning system and the fourth is for the ventilation system. There are no polarities to follow on these connectors since the relays are simply switches.

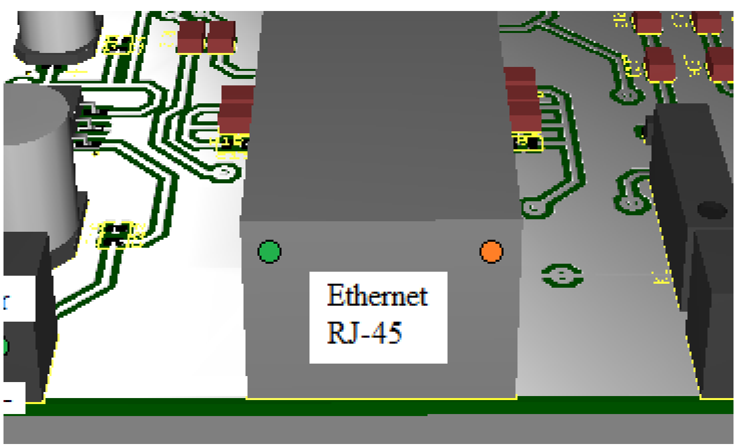
Figure 5.1.2-1: Relay Connections



5.1.3 Connecting to the network

To connect the device to the network, first connect a network cable from the RJ 45 jack on board, shown in Figure 5.1.3-1, to a free port on a router or switch.

Figure 5.1.3-1: Network Connections



Once the connection is done, the green LED on the Ethernet jack should be turned on. The orange LED turns on when the interface receives data and might not blink right away.

5.1.4 Network connection to device

Initially, the device is set on the 192.168.2.0/24 network therefore; the user must be on this same network to connect to it using a computer. To do this, the user can connect the Internet Thermostat directly to a computer and set their network interface IP address to 192.168.2.10/255.255.255.0. With this configured, you should now be able to open a web browser and navigate to 192.168.2.20 to see the web page of the device.

5.2 Calibrating the temperature displayed

As explained above, when the Internet Thermostat first starts up, it may take a few minutes for the temperature displayed to be accurate. This is due for software compensation for board heat cause by the many components which can get quite hot during runtime.

If temperature inaccuracy persists, this may be due to the power source used. The different power source with different voltage levels make the voltage regulators, placed at the beginning of the circuit, act differently.

One way to arrange this is to calibrate the temperature compensation through the web interface by choosing the power supply value that is being used in the settings page.

If the temperature values are still inaccurate, it is possible to manually increase or decrease temperature compensation in order to calibrate the device to your specific power supply.

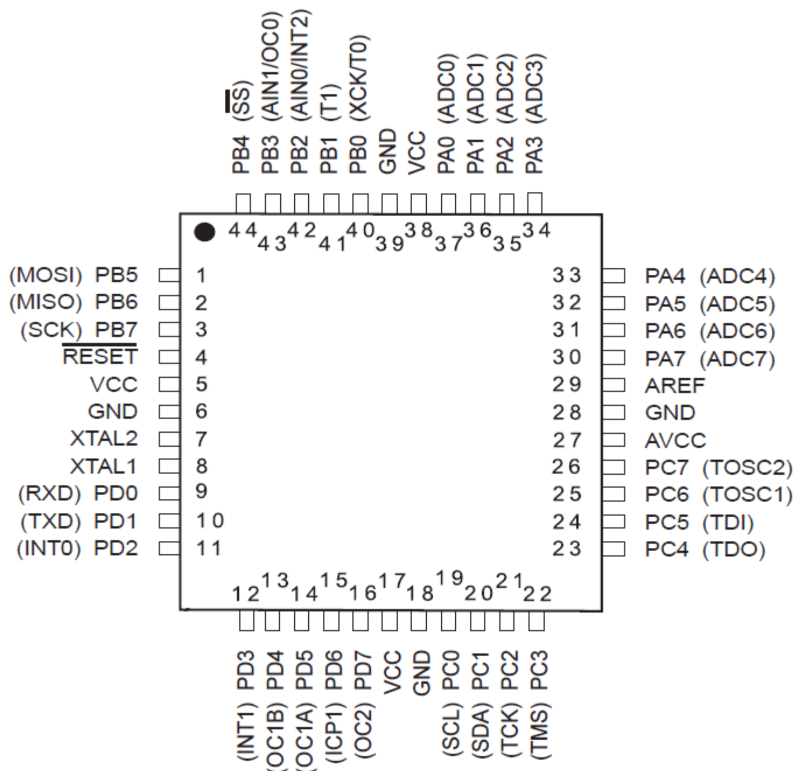
6 Technical Description

6.1 Components

6.1.1 Microcontroller

The microcontroller is the brain of the device. This Atmel ATmega32 which runs on an external 16MHz clock source contains 32KB of program memory space, 2KB of RAM, and 1KB of EEPROM. In addition to the memory space, this microcontroller offers all peripheral interfaces needed to interact with all other on board devices. These include a 10-bit ADC, an SPI interface, and 32 general I/O ports.

Figure 6.1.1-1: ATmega32 44-TQFP Package Pin Out

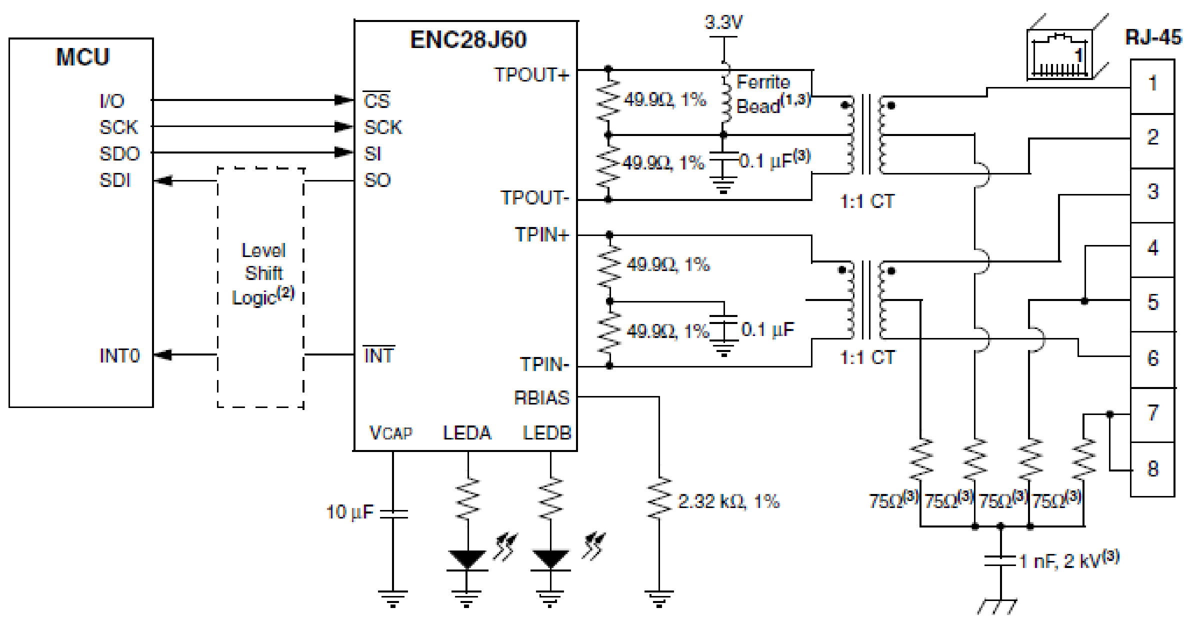


The microcontroller is programmed via a JTAG interface using the AVR Dragon. This device allows programming as well as debugging directly from AVR studio where the code is written. When the software is ready, it is compiled and imported directly in the microcontroller in a .hex file format. The final code size is roughly 20KB hence the need for a microcontroller with 32KB program memory size.

6.1.2 Ethernet Controller and Ethernet Jack

This Microchip ENC28J60 Ethernet controller is the bridge between the microcontroller and the network; it prepares full data frames and transmits those following IEEE 802.3 Ethernet standards. This device is connected through the SPI interface of the microcontroller and connects directly to the Ethernet jack using only a few external components.

Figure 6.1.2-1: ENC28J60 External Components



This device contains 8KB of buffer space which is plenty for storing received frame while they are read and to store temporary data before it is ready to transmit. The Ethernet controller is configured through the SPI by setting various configuration registers according to the desired functionality. The chip runs at 25MHz using an external crystal.

The external components needed to use this device consists of series resistors at the entry of the receive and transmit lines. These act as matching pads so that the impedance in the circuit is the same as on the transmission line. This circuit also contains filtering capacitors and a ferrite bead to act as a choke.

6.1.3 ADC and temperature sensor

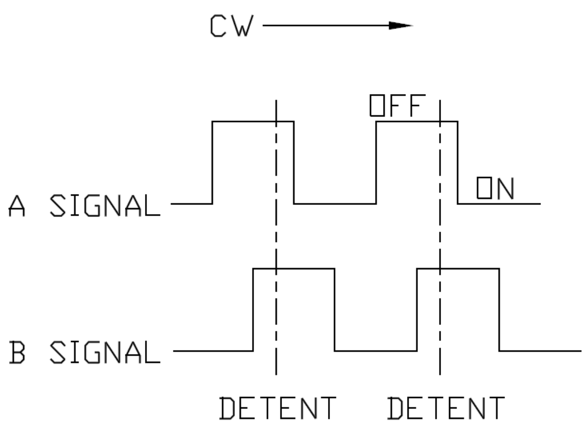
The temperature sensor used on the board is the AD22100S from Analog Devices which functions using 5V. This device simply outputs a voltage value which linearly proportional to the

temperature. Its output voltage ranges from 0V to 5V which makes it ideal for microcontroller applications. Its temperature range is from -50 degrees Celsius to 150 degrees Celsius. This means the device outputs 0V at -50 degrees Celsius and 5V at 150 degrees Celsius.
A filtering capacitor as well as a series resistor is all that is needed between the sensor and the microcontroller.

6.1.4 Rotary Encoder

This EN12 from TT Electronics is a rotary encoder which contains two internal switches. As the dial is turned left or right, the two switches turn on and off interchangeably.

Figure 6.1.4-1: Rotary Encoder Output

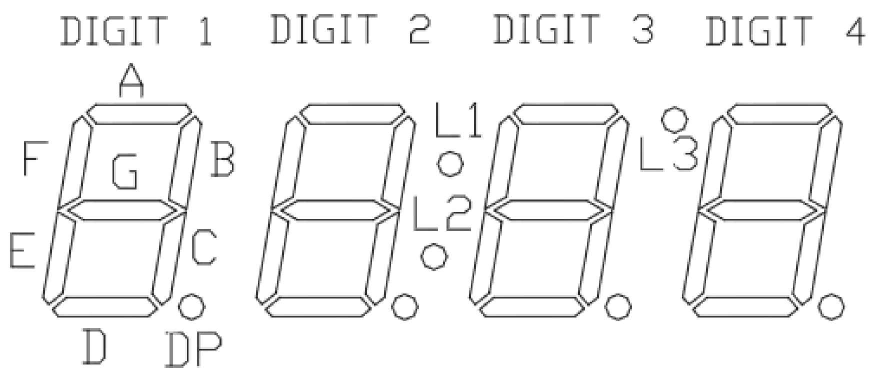


Internally, the switches both connect from each side the middle pin where they make contact when the dial is turned. To use this device, VCC was connected to each outside pin with a pull up resistor and the middle pin is connected to ground.

6.1.5 4 Digit Seven Segment Display and 74LS247

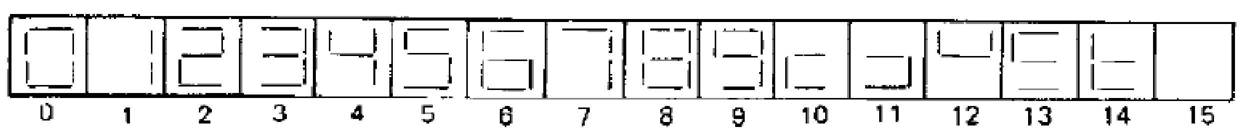
The Lite-On LTC-4727JR is a four digit seven segment display which is controlled via microcontroller general IO lines and receives data from a BCD-to-seven segment display. This display incorporates top decimal points which are a perfect fit for displaying the degrees symbol.

Figure 6.1.5-1: Different parts of the 4 Digit 7 Segment Display



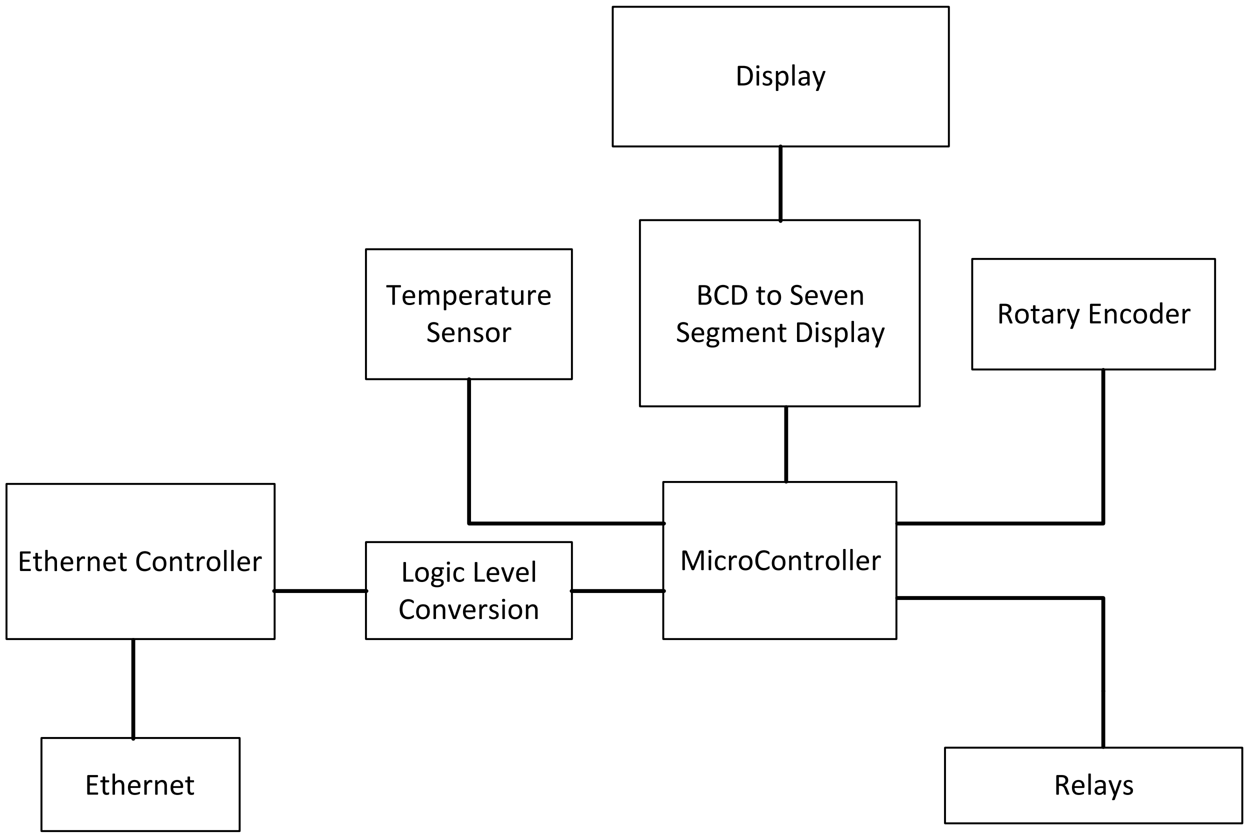
The 74LS247 was chosen for this task for its capability of sending a ‘c’ when a BCD value of 10 is sent to it. Other BCD to 7 segment ICs would not send anything to the display if the value is above 9.

Figure 6.1.5-2: 74LS247 Output Values



6.2 Interaction of Components

Figure 6.1.5-1: Interaction of Components Block Diagram



6.2.1 Main Program Tasks

In the program code, the main tasks of the microcontroller are to control the web server including all network related tasks, read the ADC data and calculate the temperature, display this value on the four digit seven segment display, verify if an external interrupt was activated from the rotary encoder, and lastly control the relays based on various reading and conditions.

Code Snippet 6.2.1-1: Main Loop

```
// loop forever
for(;;)
{
        // server process response for arp, icmp, http
        server_process ();

        // read adc value and calculate temperature
        adc0_data ();

        // display temperature values on the display
        update_display ();

        // check rotary flag to change desired temp
        check_rotary ();

        // turn on or off relays for hvac
        control_relays ();
}
```

6.2.2 Microcontroller and Networking

The microcontroller is able to communicate on the network thanks to a specific list of networking protocols that are embedded into it. These are the TCP, IP, ICMP, ARP, and HTTP protocols. Each protocol has a specific task and is very important. The Ethernet controller is also very important in this task since it is the one sending this information on the network.

The Internet Protocol is a protocol part of the network layer which is right on top of the data link layer. This part of the packet contains source and destination IP addresses. An IP address is absolutely necessary for communication on the network. In the program, the IP address is stored in external memory (EEPROM); although the Internet Thermostats current version does not allow the user to change its IP from the web interface, it would be necessary to store this IP in a non-volatile memory location in case of power shutdown or reset of the device.

Code Snippet 6.2.2-1: IP declaration in main.c

```
// set AVR ip address - stored in eeprom
BYTE ee_avr_ip[4] EEMEM = { 192, 168, 2, 20 };
```

6.2.3 Sending a Packet

The Transmission Control Protocol which is right on top of the IP protocol, according to the OSI network layer model, controls the whole receive and transmit process. This part of the network segment contains different flags which represent different communication messages as well sequence and acknowledgement numbers which are used to reassemble multiple network packets.

Code Snippet 6.2.3-1: Preparing TCP packet in http.c

```
// send first page HOME
if (packet==1)
{
        dlength = http_home( rxtx_buffer );

        tcp_send_packet (
                rxtx_buffer,
                (WORD_BYTES){dest_port},
                // source port
                (WORD_BYTES){80},
                TCP_FLAG_ACK_V | TCP_FLAG_PSH_V | TCP_FLAG_FIN_V,
                // (bool)maximum segment size
                0,
                // (bool)clear sequence ack number
                0,
                // (bool)calculate new seq and seqack number
                0,
                // prepare sequence number for next packet
                1,
                // tcp data length
                dlength,
                // server mac address
                dest_mac,
                // server ip address
                dest_ip );
}
```

The Address Resolution Protocol is used to translate an IP address into a MAC address which is also an absolute when communicating on the network. This protocol is a link layer protocol and is never used in internetwork situations. This is only used inside a local network. When a host has data to send on the network to an IP address, first an ARP frame is sent in order to translate this IP address to a MAC address and then the data can be sent.

The Internet Control Message Protocol is also a very important protocol. It is used for various operations including failed data transmission.

The general network process of the board is that it listens for incoming demands including TCP SYN, ACKs, or FINACKs, HTTP Get or Post requests, ARP requests or ICMP pings and responds accordingly.

When sending out a web page, the microcontroller prepares all the frame sections including the IP, TCP and data section and sends it to a global variable called rxtx_buffer. This buffer stores all data before it is sent out on the network. Once the packet is ready for transmission, it is sent out on the ENC28J60 buffer which then takes care of transmitting it.

Code Snippet 6.2.3-2: Preparing HTML in RXTX buffer

```
dlen = tcp_puts_data_p ( rxtx_buffer, PSTR ( "<button type='submit'name='?sub'value='1?'>LED1</button>"), dlen );
dlen = tcp_puts_data_p ( rxtx_buffer, PSTR ( "<button type='submit'name='?sub'value='2?'>LED2</button></FORM>" ), dlen );
```

The Ethernet controller takes care of the Ethernet Preamble including source and destination MAC addresses; these are loaded up into data registers. Therefore, the microcontroller only needs to send the IP, TCP and data and the Ethernet controller takes care of the rest.

6.2.4 Receiving a Packet

When receiving a packet, the Internet Thermostat functions about the same as sending it but in reverse order. In this case, it is the Ethernet Controller that sends the data to the microcontroller and this gets stored in the rxtx_buffer. Whenever a packet is received, throughout the server process, this packet is observed to see what kind of packet this is. If this is a packet with the SYN flag enabled, then the micro will then know to send a simple SYNACK packet.

Code Snippet 6.2.4-1: Sending SYNACK if SYN is received

```
// received packet with flags "SYN", let's send "SYNACK"
if ( (rxtx_buffer[ TCP_FLAGS_P ] & TCP_FLAG_SYN_V) )
{
        tcp_send_packet (
                rxtx_buffer,
                (WORD_BYTES){dest_port},
                (WORD_BYTES){80},                       // source port
                TCP_FLAG_SYN_V|TCP_FLAG_ACK_V,          // flag
                1,                                      // (bool)maximum segment size
                0,                                      // (bool)clear sequence ack number
                1,                                      // (bool)calculate new seq and seqack number
                0,                                      // tcp data length
                0,
                dest_mac,                               // server mac address
                dest_ip );                              // server ip address
        return;
}
```

The process is similar for FINACK packets received. If these flags are not on in the rxtx_buffer when a packet is received, then either the packet is an HTTP GET request for the web page or a POST request from the web page to change settings or page. If no POST is detected in the buffer, then the program will continue on and send the web page. If POST is detected, then further analysis of the buffer is performed to go verify what the POST data is. Depending on various different names and values sent, different tasks are done such as turn on or off the relays, increase or decrease the temperature, or calibrate the temperature compensation. The value attached to the name of the POST data is found and stored in the generic buffer where they can be checked later. The POST data is sent in the form: ?%sub=1%?. The percent and question mark symbol make the data easier to find throughout packets that often vary greatly in length depending on browsers or device.

Code Snippet 6.2.4-2: Observing POST request from received packet

```
else if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fsub" ), generic_buf ) )
{
        post_data=0;

        if (mac_filter_pass==1)
        {
                if (generic_buf[0]==49)
                        PORTB ^= 0x02;
                else if (generic_buf[0]==50)
                        PORTB ^= 0x01;

                else if (generic_buf[0]==117)
                {
                        desired_temp += 0.1;
                        TCNT1 = 0;
                        rot_enc_change=1;
                }

                else if (generic_buf[0]==100)
                {
                        desired_temp -= 0.1;
                        TCNT1 = 0;
                        rot_enc_change=1;
                }
        }
}
```

6.2.5 Web Page

The web pages sent by the microcontroller contain a number of features that are important to consider. The tabs layout that gives the options of Home and Settings are CSS styles that are defined in the html. The HREF links were replaced with POST method in order to always keep the URL area the same address and to only have to use one method of verifying sent data. Other CSS styles were defined so that the page would be centered and would adjust to mobile phones.

6.2.6 Rotary Encoder and Display

Both outputs, A and B of the rotary encoder are connected to an AND gate which outputs a logic LOW whenever both outputs are not HIGH at the same time. It can be observed that whenever the dial is turned clockwise, output A is the first to go LOW right after they are both HIGH and that when the dial is turned counter clockwise; B is the first to go LOW. Pin B is connected to a general IO line of the controller. The output of this AND gate is connected to external interrupt INT2 of the microcontroller and an interrupt sub routine is triggered on the falling edge of this pin. In this interrupt routine, a flag is set to tell the program to then verify if pin B of the dial is LOW or HIGH. This will determine if the dial is turned clockwise or counter clockwise. Also, two counters are reset to 0. One of the counters is used to count how long the desired temperature is displayed on the display and the other one is used to count how long to wait before verifying the rotary encoder input; there is a little noise which can cause instability when rotating the dial.

Code Snippet 6.2.6-1: Rotary Encoder Interrupt Sub Routine

```
ISR (INT2_vect)                    //int2 interrupt subroutine
{
            if (rot_enc_flag==0)
            {
                        TCNT0 = 0;
                        TCNT1 = 0;
                        rot_enc_flag = 1;
                        rot_enc_change = 1;
            }
}
```

In every loop of the main program, a function is called to update the display. If the flag for the rotary encoder interrupt is set, then the value being changed is displayed on the display. A counter within the microcontroller counts five seconds before going back to displaying the current temperature. Within these five seconds, if the interrupt is triggered again, then the count is restarted; if not, then the software interrupt flag is cleared. Once cleared, the display goes back to displaying the current temperature.

6.2.7 Temperature Sensor and Relays

In the program, a global float value of the current and desired temperature values is always kept and used for various functions. The ADC is configured in a way that a function is called every time to take a sample of the ADC value. This 10 bit sample is then added to a 32 bit value 4096 times and then divided by 4096 in order to get the average. This process takes about 5 seconds to complete and gives smoother and more accurate temperature readings.

Code Snippet 6.2.7-1: ADC average and conversion

```
// Read ADCL before ADCH
adc0_sum += ((ADCL) | ((ADCH)<<8));

if(x==4095)
{
        adc0_average = adc0_sum >> 12;
        adc0_temp = (adc0_average*0.2172)-61.07;    //equation for AD22100
        adc0_temp -= adc0_heat_compensation;        //To compensate for board heat
        adc0_sum = 0;
        x=0;
}
else
        x++;
```

In the main loop is a function to control the relays. This function will subtract the current temperature value by the desired temperature value and start the relays according to the differences. Also, the web interface Settings page includes many features to have complete control over relays. These settings are also verified before activating or closing anything. Flags are set depending on different conditions and then different parts of the system are activated or deactivated according to which flags are set or not.

Code Snippet 6.2.7-2: Relay flags being set according to conditions

```
        if (temp_difference<=-1)
        {
                relays_cnt0++;
                if (relays_cnt0==cnt0end)
                {
                        relays_cnt0=0;
                        hvac_heat=1;
                        hvac_ac=0;
                        hvac_vent=0;
                }
        }
```

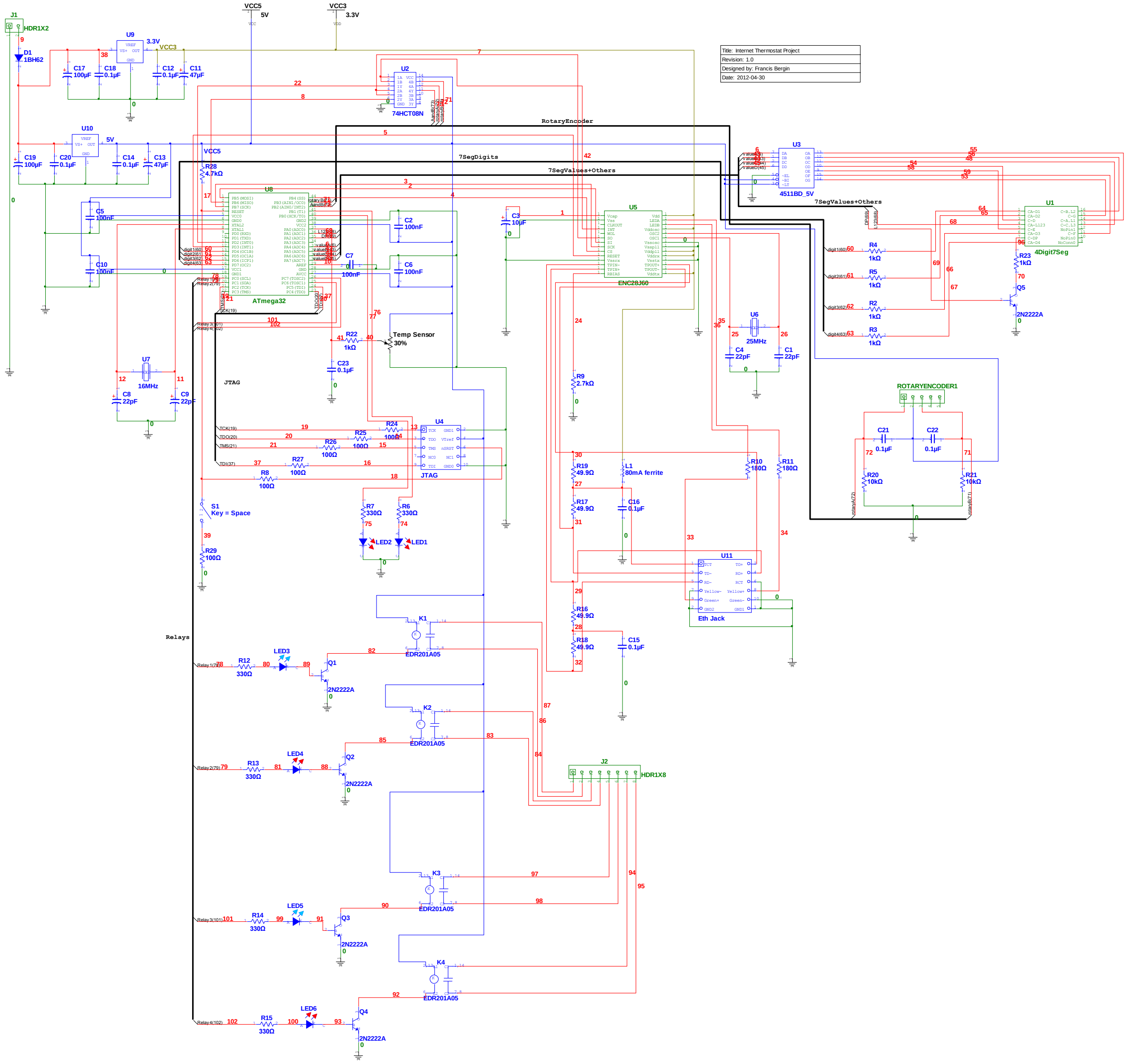
Code Snippet 6.2.7-3: System control based on set flags

```
if (((hvac_vent_flag==1)&&(hvac_vent==1))||((hvac_vent_flag==1)&&(hvac_flag_auto==0)))
{
        PORTC &= ~(1 << 0);
        PORTC &= ~(1 << 1);
        PORTC &= ~(1 << 6);
        PORTC |= 1 << 7;
        return;
}
```

7 Appendix

7.1 Schematics

Figure 6.2.7-1: Design Schematic



7.2 PCB Layout

Figure 6.2.7-1: PCB Layout: Top

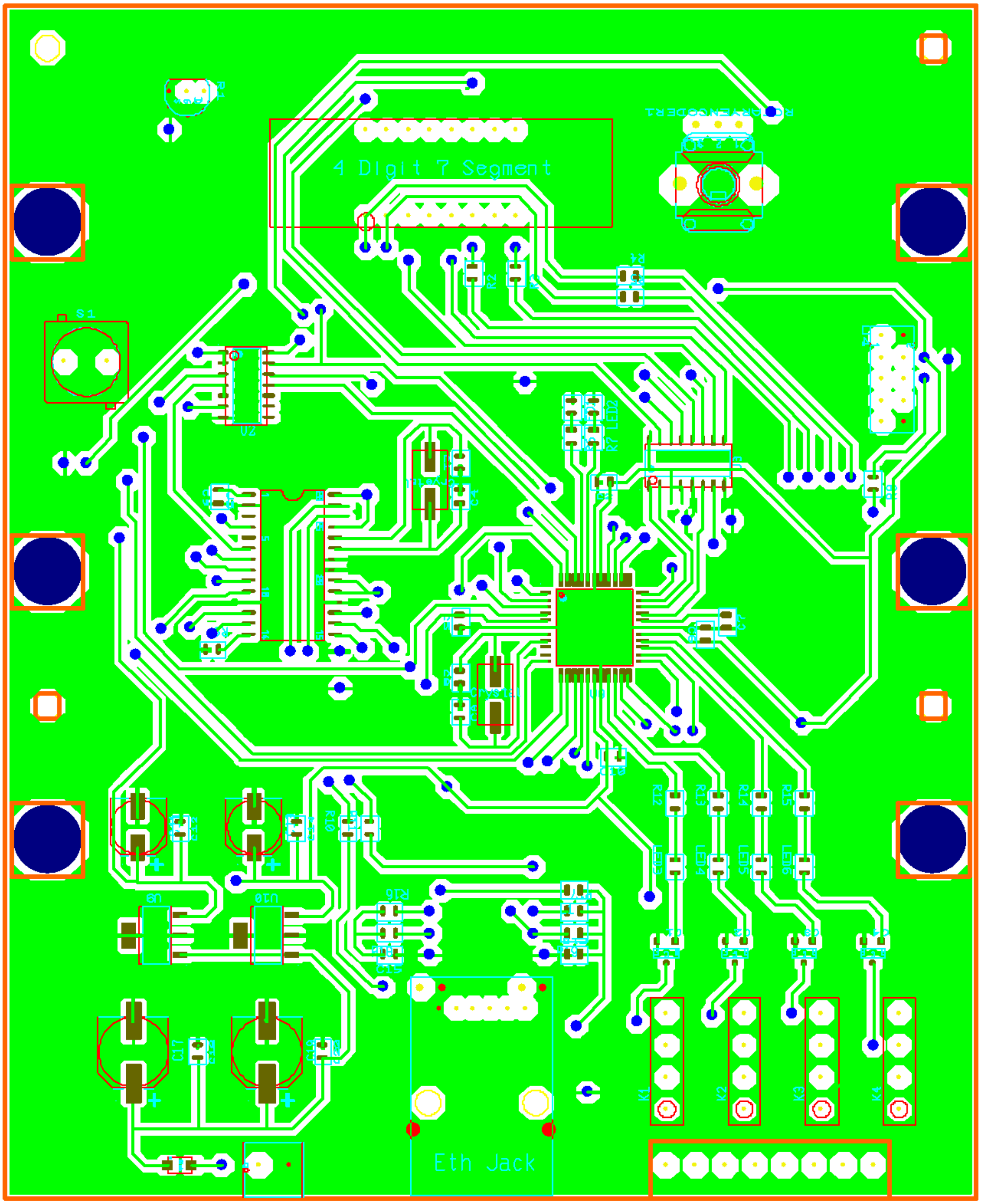
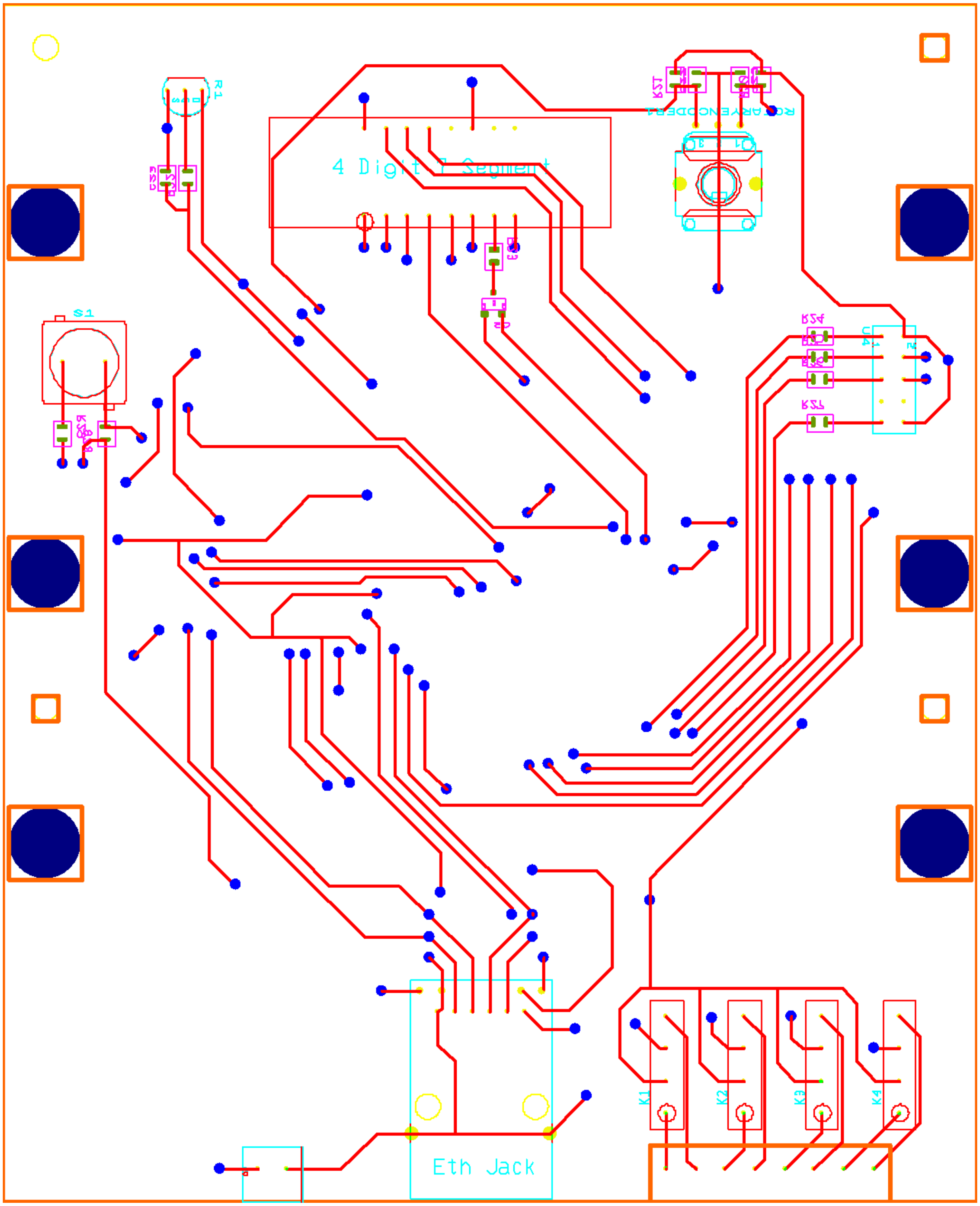


Figure 6.2.7-2: PCB Layout: Bottom



7.3 Parts List / Bill of Materials

Table 7.3-1: Bill of Materials

REFDES	QTY.	VALUE	SHAPE
C12,C14,C15,C16,C18,C20,C21,C22,C23	9	0.1uF	C0603S
D1	1	1BH62	SMF-Diode
R2,R3,R4,R5,R22,R23	6	1kOhm	R0603S
R1	1	1kOhm	TO-92(94)
L1	1	1mH	R0603S
R9	1	2.7kOhm	R0603S
Q1,Q2,Q3,Q4,Q5	5	2N2222A	TRA_SOT23S
R28	1	4.7kOhm	R0603S
U1	1	4Digit7Seg	Lite-On 4 digit 7 segment
R20,R21	2	10kOhm	R0603S
C3	1	10uF	C0603S
C1,C4,C8,C9	4	22pF	C0603S
C11,C13	2	47uF	CAPAE660X540N
R16,R17,R18,R19	4	49.9Ohm	R0603S
U2	1	74LS08J	SOIC127P600X175-14N
C2,C5,C6,C7,C10	5	100nF	C0603S
C17,C19	2	100uF	CAPAE830X1020N
R8,R24,R25,R26,R27,R29	6	100Ohm	R0603S
R10,R11	2	180Ohm	R0603S
R6,R7,R12,R13,R14,R15	6	330Ohm	R0603S
U3	1	4511BD_5V	SOIC127P600X175-16N
U8	1	ATmega32	TQFP80P1200X1200X160-44N
U6,U7	2	CRYSTAL_VIRTUAL	HC-49USX Crystal
U4	1	DragonJTAG	HDR2X5
K1,K2,K3,K4	4	EDR201A05	MS05-Reed-Relay
U5	1	ENC28J60	SO28
J1	1	HDR1X2	ED555/2DS
ROTARYENCODER1	1	HDR1X5	P_RK09K113A4
J2	1	HDR1X8	ED555/8DS
LED3,LED5	2	LED_blue	R0603S
LED1,LED2,LED4,LED6	4	LED_red	R0603S
U11	1	MagJackEthernet	Eth-507-1445
U9	1	NCP1117DT33G	SOT230P700X180-4N
U10	1	NCP1117DT50G	SOT230P700X180-4N
S1	1	SPST	TAS_11X9R5

7.4 Cost Analysis

Estimated cost per unit

Internet Thermostat Board:

- Components **125.00**
- Copper sheet **15.00**
- Solder & flux **15.00**

Time – milling, soldering:

- 10Hrs **175.00**

Non-Recurring Engineering

Materials:

- JTAG debugger **55.00**

Time – research, planning:

- 15 Weeks ≈ 20Hrs/Week **5,250.00**

Total **5,635.00$ CAD**

7.5 Source Code Listing

7.5.1 main.c

Code Snippet 7.5.1-1: main functions in main.c

```
int main (void)
{
	// change your mac address here
	avr_mac.byte[0] = 0x00;
	avr_mac.byte[1] = 0x04;
	avr_mac.byte[2] = 0xa3;
	avr_mac.byte[3] = 0xb7;
	avr_mac.byte[4] = 0x73;
	avr_mac.byte[5] = 0x6f;

	// read avr and server ip from eeprom
	eeprom_read_block ( &avr_ip, ee_avr_ip, 4 );

	// setup clock for timer0 and timer 1
	TCCR1B = 0x05;
	TCCR0 = 0x05;

	// initial enc28j60
	enc28j60_init( (BYTE*)&avr_mac );

	// initialize adc
	adc0_init();

	// initialize I/O
	DDRA = DDRA   | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 4)
	              | (1 << 5) | (1 << 6) | (1 << 7);
	DDRB = DDRB | (1 << 0) | (1 << 1);
	DDRC = DDRC | (1 << 0) | (1 << 1) | (1 << 7) | (1 << 8);
	DDRD = DDRD | (1 << 4) | (1 << 5) | (1 << 6) | (1 << 7);

	// initialize interupt for rotary encoder
	DDRB &= ~(1 << 3);
	GICR |= (1<<INT2);
	MCUCSR |= (0<<ISC2);
	sei();

	// loop forever
	for(;;)
	{
		// server process response for arp, icmp, http
		server_process ();

		// read adc value and calculate temperature
		adc0_data ();

		// display temperature values on the display
		update_display ();

		// check rotary flag to change desired temp
		check_rotary ();

		// turn on or off relays for hvac
		control_relays ();
	}

	return 0;
}
```

7.5.2 adc.c

Code Snippet 7.5.2-1: variables in adc.c

```
#include "includes.h"

DWORD adc0_average;
DWORD adc0_sum;
WORD adc0_check;            // Does not need to be global
WORD x;

WORD temp_value1;
WORD temp_value2;
WORD temp_value3;

BYTE adc0_valid=0;          // to wait until first conversion in done

float adc0_temp;

float adc0_calc_slope = (200.0/921.6);
float adc0_calc_intercept = (550.0/9.0);

float adc0_heat_compensation=15.0;
float adc0_heat_comp_custom=5.0;

BYTE adc0_pwr_src=1;

void adc0_init ( void )
{
       ADCSRA = 0x80;
       ADMUX = 0x40;
}

WORD adc0_read (void)
{
       return adc0_average;
}
```

Code Snippet 7.5.2-2: adc0_data function

```
void adc0_data (void)
{

	// Start conversion
	ADCSRA |= (1 << ADSC);

	// Wait until conversion complete
	while ( !(ADCSRA & (1<<ADIF)))
		;

	// CAUTION: READ ADCL BEFORE ADCH!!!
	adc0_check = ((ADCL) | ((ADCH)<<8));
	adc0_sum += ((ADCL) | ((ADCH)<<8));

	if (adc0_valid==0)
	{
		if(x==511)
		{
			adc0_valid=1;

			adc0_average = adc0_sum >> 9;
			adc0_temp = (adc0_average*adc0_calc_slope)-adc0_calc_intercept;
	//equation according to AD22100 characteristics
			adc0_temp -= adc0_heat_compensation;
	//To compensate for board heat
			adc0_sum = 0;
			x=0;
		}
		else
			x++;
	}

	if (adc0_valid==1)
	{
		if(x==4095)
		{
			adc0_average = adc0_sum >> 12;
			adc0_temp = (adc0_average*0.2172)-61.07;  //equation according to
	AD22100 characteristics
			adc0_temp -= adc0_heat_compensation;
		//To compensate for board heat
			adc0_sum = 0;
			x=0;
		}
		else
			x++;
	}
}
```

Code Snippet 7.5.2-3: temperature value functions

```
WORD adc_read_temp1 ( void )
{
	temp_value1 = adc0_temp/10;
	return temp_value1;
}

WORD adc_read_temp2 ( void )
{
	WORD temp0, temp1;

	temp0 = adc0_temp/10;
	temp1 = temp0 * 10;
	temp_value2 = adc0_temp - temp1;

	return temp_value2;
}

WORD adc_read_temp3 ( void )
{
	WORD temp0, temp1;

	temp0 = adc0_temp;
	temp1 = temp0 * 10;
	temp_value3 = (adc0_temp*10) - temp1;

	return temp_value3;
}
```

Code Snippet 7.5.2-4: temperature compensation values

```
WORD adc_read_compensation1 ( void )
{
	temp_value1 = adc0_heat_compensation/10;
	return temp_value1;
}

WORD adc_read_compensation2 ( void )
{
	WORD temp0, temp1;

	temp0 = adc0_heat_compensation/10;
	temp1 = temp0 * 10;
	temp_value2 = adc0_heat_compensation - temp1;

	return temp_value2;
}

WORD adc_read_compensation3 ( void )
{
	WORD temp0, temp1;

	temp0 = adc0_heat_compensation;
	temp1 = temp0 * 10;
	temp_value3 = (adc0_heat_compensation*10) - temp1;

	return temp_value3;
}
```

7.5.3 display.c

Code Snippet 7.5.3-1: display.c display null or desired temperature

```
void update_display ( void )
{
	WORD disp_value1, disp_value2, disp_value3;

	DDRA = DDRA | 0xf0;
	DDRD = DDRD | 0x70;


	if(adc0_valid==0)
	{
	//everything is on
		PORTD |= (1 << 4);
		PORTD |= (1 << 5);
		PORTD |= (1 << 6);
		PORTD |= (1 << 7);
		PORTA = 8*16;
		PORTA |= (1 << 2);
		PORTA |= (1 << 1);
		_delay_ms(1);
	}

	if(rot_enc_change==1)
	{
		disp_value1 = desired_read_temp1();
		disp_value2 = desired_read_temp2();
		disp_value3 = desired_read_temp3();

	//first digit
		PORTD |= (1 << 4);
		PORTD &= ~(1 << 5);
		PORTD &= ~(1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value1*16;
		PORTA &= ~(1 << 2);
		_delay_ms(1);


	//second digit + decimal point
		PORTD &= ~(1 << 4);
		PORTD |= (1 << 5);
		PORTD &= ~(1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value2*16;
		PORTA |= (1 << 2);
		_delay_ms(1);

	//third digit
		PORTD &= ~(1 << 4);
		PORTD &= ~(1 << 5);
		PORTD |= (1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value3*16;
		PORTA &= ~(1 << 2);
		_delay_ms(1);
	}
```

Code Snippet 7.5.3-2: display.c display current temperature

```
else
	{
		disp_value1 = adc_read_temp1();
		disp_value2 = adc_read_temp2();
		disp_value3 = adc_read_temp3();


	//first digit
		PORTD |= (1 << 4);
		PORTD &= ~(1 << 5);
		PORTD &= ~(1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value1*16;
		PORTA &= ~(1 << 2);
		_delay_ms(1);

	//second digit + decimal point
		PORTD &= ~(1 << 4);
		PORTD |= (1 << 5);
		PORTD &= ~(1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value2*16;
		PORTA |= (1 << 2);
		_delay_ms(1);

	//third digit
		PORTD &= ~(1 << 4);
		PORTD &= ~(1 << 5);
		PORTD |= (1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = disp_value3*16;
		PORTA &= ~(1 << 2);
		_delay_ms(1);

	//c at the end
		PORTD &= ~(1 << 4);
		PORTD &= ~(1 << 5);
		PORTD &= ~(1 << 6);
		PORTD |= (1 << 7);
		PORTA = 10*16;
		PORTA &= ~(1 << 2);
		_delay_ms(1);

	//dot over second digit
		PORTD &= ~(1 << 4);
		PORTD &= ~(1 << 5);
		PORTD &= ~(1 << 6);
		PORTD &= ~(1 << 7);
		PORTA = 11*16;
		PORTA &= ~(1 << 2);
		PORTA |= (1 << 1);
		_delay_ms(1);

	}
}
```

7.5.4 rotaryencoder.c

Code Snippet 7.5.4-1: rotary encoder interrupt and check rotary functions

```
#include "includes.h"

//global variables
BYTE rot_enc_flag;
BYTE rot_enc_change=0;

float desired_temp=20;

ISR (INT2_vect)             //int2 interrupt subroutine
{
	if (rot_enc_flag==0)
	{
		TCNT0 = 0;
		TCNT1 = 0;
		rot_enc_flag = 1;
		rot_enc_change = 1;
	}
}


void check_rotary (void)
{
	int input;

	if (rot_enc_flag==1)
	{
		if (TCNT0 > 16)
		{
			input = PINB;

			if ((input|0xFB)==0xFB)
			{
				if ((input|0xF7)==0xF7)
				{
					//PORTB ^= 0x01;
					desired_temp -= 0.1;
				}

				else if ((input|0xF7)==0xFF)
				{
					//PORTB ^= 0x02;
					desired_temp += 0.1;
				}

				rot_enc_flag = 0;
			}
		}
	}

	if(TCNT1 > 45000)
	{
		rot_enc_change=0;
		TCNT1 = 0;
	}
}
```

Code Snippet 7.5.4-2: desired temperature values

```
WORD desired_read_temp1 ( void )
{
	WORD temp_value1;
	temp_value1 = desired_temp/10;
	return temp_value1;
}

WORD desired_read_temp2 ( void )
{
	WORD temp_value2;
	WORD temp0, temp1;

	temp0 = desired_temp/10;
	temp1 = temp0 * 10;
	temp_value2 = desired_temp - temp1;

	return temp_value2;
}

WORD desired_read_temp3 ( void )
{
	WORD temp_value3;
	WORD temp0, temp1;

	temp0 = desired_temp;
	temp1 = temp0 * 10;
	temp_value3 = (desired_temp*10) - temp1;

	return temp_value3;
}
```

7.5.5 relays.c

Code Snippet 7.5.5-1: relays.c code 1

```
WORD cnt0end=1500;
WORD relays_cnt0=1499;

float temp_difference;

// overall hvac system on or off
BYTE hvac_flag_sys=1;
BYTE hvac_flag_auto=1;
BYTE hvac_flag_heat_src=1;

// hvac components on or off
BYTE hvac_heat_flag=1, hvac_ac_flag=1, hvac_vent_flag=1;

BYTE hvac_heat=0, hvac_ac=0, hvac_vent=0;


void control_relays (void)
{
	if (adc0_valid==0)
	{
		//PORTC |= 0xc3;
		return;
	}

	if (hvac_flag_sys==0)
	{
		PORTC &= ~(1 << 0);
		PORTC &= ~(1 << 1);
		PORTC &= ~(1 << 6);
		PORTC &= ~(1 << 7);
		relays_cnt0=0;
		return;
	}

	if (rot_enc_change==1)
		return;


	temp_difference=adc0_temp-desired_temp;

	if (temp_difference<=-1)
	{
		relays_cnt0++;
		if (relays_cnt0==cnt0end)
		{
			relays_cnt0=0;
			hvac_heat=1;
			hvac_ac=0;
			hvac_vent=0;
		}
	}
```

Code Snippet 7.5.5-2: relays.c code 2

```
	else
	{
		hvac_heat=0;
		hvac_ac=0;
		hvac_vent=0;
	}


	if
(((hvac_heat_flag==1)&&(hvac_heat==1))||((hvac_heat_flag==1)&&(hvac_flag_auto==0)))
	{
		if (hvac_flag_heat_src==1)
		{
			PORTC &= ~(1 << 0);
			PORTC |= 1 << 1;
			PORTC &= ~(1 << 6);
			PORTC |= 1 << 7;
			return;
		}

		else if (hvac_flag_heat_src==2)
		{
			PORTC |= 1 << 0;
			PORTC &= ~(1 << 1);
			PORTC &= ~(1 << 6);
			PORTC &= ~(1 << 7);
			return;
		}
	}
```

Code Snippet 7.5.5-3: relays.c code 3

```
	if (((hvac_ac_flag==1)&&(hvac_ac==1))||((hvac_ac_flag==1)&&(hvac_flag_auto==0)))
	{
		PORTC &= ~(1 << 0);
		PORTC &= ~(1 << 1);
		PORTC |= 1 << 6;
		PORTC |= 1 << 7;
		return;
	}

	if
(((hvac_vent_flag==1)&&(hvac_vent==1))||((hvac_vent_flag==1)&&(hvac_flag_auto==0)))
	{
		PORTC &= ~(1 << 0);
		PORTC &= ~(1 << 1);
		PORTC &= ~(1 << 6);
		PORTC |= 1 << 7;
		return;
	}

	PORTC &= ~(1 << 0);
	PORTC &= ~(1 << 1);
	PORTC &= ~(1 << 6);
	PORTC &= ~(1 << 7);

}
```

7.5.6 http.c

Code Snippet 7.5.6-1: http.c verifying post and mac address

```
if ((http_get_post ( rxtx_buffer, dlength, PSTR( "OST " ), generic_buf )) ||
(post_data==1))
{


	if (mac_filter==0)
	{
		mac_filter_pass=1;
	}
	else if (mac_filter==1)
	{
		if
((rxtx_buffer[9]==mac_filter_address[0])&&(rxtx_buffer[10]==mac_filter_address[1])&&(rxtx
_buffer[11]==mac_filter_address[2]))
		{
			mac_filter_pass=1;
		}
		else
			mac_filter_pass=0;

	}


	if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fpage" ), generic_buf ) )
	{
		post_data=0;

		if (generic_buf[0]==49)
			web_page=1;
		else if (generic_buf[0]==50)
			web_page=2;
	}


	else if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fmac" ), generic_buf )
)
	{
		post_data=0;

		if (mac_filter_pass==1)
		{
			if (generic_buf[0]==48)
				mac_filter = 0;
			else if (generic_buf[0]==49)
			{
				mac_filter = 1;


				mac_filter_address[0]=rxtx_buffer[9];

				mac_filter_address[1]=rxtx_buffer[10];

				mac_filter_address[2]=rxtx_buffer[11];
			}

		}
	}
```

Code Snippet 7.5.6-2: http.c verifying post data

```
else if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fsub" ), generic_buf ) )
{
	post_data=0;

	if (mac_filter_pass==1)
	{
		if (generic_buf[0]==49)
			PORTB ^= 0x02;
		else if (generic_buf[0]==50)
			PORTB ^= 0x01;

		else if (generic_buf[0]==117)
		{
			desired_temp += 0.1;
			TCNT1 = 0;
			rot_enc_change=1;
		}

		else if (generic_buf[0]==100)
		{
			desired_temp -= 0.1;
			TCNT1 = 0;
			rot_enc_change=1;
		}
	}
}


else if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fhvac" ), generic_buf ) )
{
	post_data=0;

	if (mac_filter_pass==1)
	{
		if (generic_buf[0]==48)
			hvac_flag_sys ^= 0x01;

		else if (generic_buf[0]==49)
			hvac_heat_flag ^= 0x01;

		else if (generic_buf[0]==50)
			hvac_vent_flag ^= 0x01;

		else if (generic_buf[0]==51)
			hvac_ac_flag ^= 0x01;


		else if (generic_buf[0]==112)
			hvac_flag_heat_src = 1;

		else if (generic_buf[0]==115)
			hvac_flag_heat_src = 2;
```

Code Snippet 7.5.6-3: http.c verifying post data

```
else if (generic_buf[0]==97)
	{
		hvac_flag_auto = 1;
		hvac_heat_flag = 1;
		hvac_vent_flag = 1;
		hvac_ac_flag = 1;
	}
		else if (generic_buf[0]==109)
			{
				hvac_flag_auto = 0;
				hvac_heat_flag = 0;
				hvac_vent_flag = 0;
				hvac_ac_flag = 0;
			}

			else if (generic_buf[0]==120)
				;
		}
	}
else if ( http_get_variable ( rxtx_buffer, dlength, PSTR( "3Fadcsrc" ), generic_buf ) )
	{
		post_data=0;

		if (mac_filter_pass==1)
		{
			if (generic_buf[0]==48)
			{	adc0_pwr_src=0;
				adc0_heat_compensation=5.0;
			}
			if (generic_buf[0]==49)
			{	adc0_pwr_src=1;
				adc0_heat_compensation=15.0;
			}
			else if (generic_buf[0]==50)
			{	adc0_pwr_src=2;
				adc0_heat_compensation=17.5;
			}
			else if (generic_buf[0]==51)
			{	adc0_pwr_src=3;
				adc0_heat_compensation=20.0;
			}
			//heat compensation up or down
			else if (generic_buf[0]==117)
			{	adc0_pwr_src=0;
				adc0_heat_compensation += 0.1;
			}
			else if (generic_buf[0]==100)
			{	adc0_pwr_src=0;
				adc0_heat_compensation -= 0.1;
			}
		}
	}

	else if (post_data==0)
	{	post_data=1;
		return;
	}
}
```

Code Snippet 7.5.6-4: http.c get_post function

```
BYTE http_get_post ( BYTE *rxtx_buffer, WORD dlength, PGM_P val_key, BYTE *dest )
{
	WORD data_p; //data_p_test;
	PGM_P key;
	BYTE match=0, temp;
	//WORD sent_value;
	//WORD clength;

	key = val_key;

	if ((rxtx_buffer[0x36]==0x50)&&(rxtx_buffer[0x39]==0x54))
		dlength+=60;

	// get data position
	data_p = tcp_get_hlength( rxtx_buffer ) + sizeof(ETH_HEADER) + sizeof(IP_HEADER);

// Find '?' in rx buffer, if found '?' in rx buffer then let's find variable key(val_key)
	for ( ; data_p<dlength; data_p++ )
	{
		if ( rxtx_buffer [ data_p ] == 'P' )
			break;
	}
	// not found '?' in buffer
	if ( data_p == dlength )
		return 0;

	// find variable key in buffer
	for ( ; data_p<dlength; data_p++ )
	{
		temp = pgm_read_byte ( key );

		// end of variable keyword
		if ( rxtx_buffer [ data_p ] == '/' && match != 0 )
		{
			if ( temp == '\0' )
			{
				//return match;
				data_p++;
				break;
			}
		}
		// variable keyword match with rx buffer
		if ( rxtx_buffer [ data_p ] == temp )
		{
			key++;
			match++;
		}
		else
		{
			// no match in rx buffer reset match and find again
			key = val_key;
			match = 0;
		}
	}

	return match;
}
```

7.5.7 tcp.c

Code Snippet 7.5.7-1: tcp.c noting of previous sequence number

```
void tcp_send_packet (
	BYTE *rxtx_buffer,
	WORD_BYTES dest_port,
	WORD_BYTES src_port,
	BYTE flags,
	BYTE max_segment_size,
	BYTE clear_seqack,
	WORD next_ack_num,
	WORD next_seq_num,
	WORD dlength,
	BYTE *dest_mac,
	BYTE *dest_ip )
{
	BYTE i, tseq;
	WORD_BYTES ck;

	// generate ethernet header
	eth_generate_header ( rxtx_buffer, (WORD_BYTES){ETH_TYPE_IP_V}, dest_mac );

	// sequence numbers:
	// add the rel ack num to SEQACK
	if ( next_ack_num )
	{
		for( i=4; i>0; i-- )
		{
			next_ack_num = rxtx_buffer [ TCP_SEQ_P + i - 1] + next_ack_num;
			tseq = rxtx_buffer [ TCP_SEQACK_P + i - 1];
			rxtx_buffer [ TCP_SEQACK_P + i - 1] = 0xff & next_ack_num;

			// copy the acknum sent to us into the sequence number
			rxtx_buffer[ TCP_SEQ_P + i - 1 ] = tseq;
			prev_seq_number[i] = 0xff & next_ack_num;

			next_ack_num >>= 8;
		}
	}
```

Code Snippet 7.5.7-2: tcp.c calculation of next sequence number

```
	if ( next_seq_num )
	{
		if (packet==0)
		{
			prev_dlength0=dlength;
		}

		else if (packet==1)
		{
			prev_dlength=prev_dlength0;
		}

		else if (packet==2)
		{
			prev_dlength2=dlength;
			prev_dlength=prev_dlength0;
		}

		else if (packet==3)
		{
			prev_dlength=prev_dlength2;
		}


		if ((packet==1)||(packet==2)||(packet==3))
		{
			prev_cnt++;
			y=7;
			for (cntx=0; cntx<=3; cntx++)
			{
				//if (rxtx_buffer [TCP_SEQ_P + cntx ]!=0)
				//{
					prev_seq_temp = rxtx_buffer [TCP_SEQ_P + cntx ];
					prev_seq_mod = prev_seq_temp % 16;
					prev_seq_value [y-1] = prev_seq_mod;

					prev_seq_temp = prev_seq_temp / 16;
					prev_seq_mod = prev_seq_temp % 16;
					prev_seq_value [y] = prev_seq_mod;

					y-=2;
				//}
				//else
				//{
				//	y-=2;
				//}
			}
```

Code Snippet 7.5.7-3: tcp.c implementation of next sequence number

```
                powx = 16;
                prev_seq_total = 0;
                for (cnty=0; cnty<=7; cnty++)
                {
                        if(cnty==0)
                                prev_seq_total += prev_seq_value [cnty];

                        else
                        {
                                prev_seq_total += prev_seq_value [cnty] * powx;
                                powx *= 16;
                        }
                }

                next_seq_number = prev_dlength + prev_seq_total;




                //prev_cnt=0;

                y=7;
                next_seq_temp = next_seq_number;

                for (cntx=4; cntx>0; cntx--)
                {
                        next_seq_add=0;

                        next_seq_mod = next_seq_temp % 16;
                        next_seq_add = next_seq_mod;
                        next_seq_temp = next_seq_temp / 16;

                        next_seq_mod = next_seq_temp % 16;
                        next_seq_add += (next_seq_mod*16);
                        next_seq_temp = next_seq_temp / 16;

                        rxtx_buffer [TCP_SEQ_P + cntx -1 ] = next_seq_add;

                }

                //prev_cnt=0;
        }
}
```

8 GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.