```
#include <xc.h>
#include <pic.h>

	#pragma config FOSC=HS, CP=OFF, DEBUG=OFF, BORV=20, BOREN=0,
MCLRE=ON, PWRTE=ON,WDTE=OFF
	#pragma config BORSEN=OFF, IESO=OFF, FCMEN=0



#define PORTBIT(adr,bit)	((unsigned)(&adr)*8+(bit))

//--------------------
const char		T_REF_AMB = 25;  //Reference ambient temperature
const char		T_REF_PLT= 348;  //Reference plate temperature in
Kelvin
const char		T_THRESH = 5;  //Threshold Temperature Difference
const char		N_REF_AMB = 99;  // Calculated ambient adresh
reference
const char		N_REF_HOT = 123;  //Calculated hot adresh reference
const char		N_REF_COLD = 113; //Calculated cold adresh reference

static bit	toggleswitch	@	PORTBIT(PORTC,0);
	//Pointer to port C
static bit	carryBit		@	PORTBIT(STATUS,0);
	//Pointer to carry bit
bit			carry;
			//variable for carry bit
char		n_ambient;
			//adresh # for ambient
char		n_hot;
			//adresh # for hot threshold
char		n_cold;
			//adresh # for cold threshold
char		n_plt;
			//adresh value of plate reading
//-----------------

//Delay to prepare ADCON
void SetupDelay(void)
{
	for (char n =1; n > 0; n--) {}
	return;
}

// Initialization of Ports
void	init(void)
{
	PORTD = 0B00000000;
	TRISD = 0B00000000;
	PORTB = 0B00000000;
	TRISB = 0B00000000;
	PORTA = 0B00000000;
	TRISA = 0B00111111;
```

```
        PORTC = 0B00000001;
        TRISC = 0B00111111;
        ADCON1 = 0b00000100;
        ADCON0 = 0b01001001;
        SetupDelay();
        GO     = 1;
        return;
}

//Computes Reference Values
void get_n_values(void)
{
        /*ADCON0 = 0b01000001; //AN1, sets GO bit
        GO = 1;
        while(GO == 1){}
        while(GO == 1){}
        char n_amb_1 = ADRESH;

        ADCON0 = 0b01000001; //AN1, sets GO bit
        GO = 1;
        while(GO ==1){}
        while(GO ==1){}
        char n_amb_2 = ADRESH;*/

        // Reads plate thermocouple
        ADCON0 = 0b01001001; //AN0, sets GO bit
        GO = 1;
        while(GO == 1){}
        while(GO == 1){}
        char n_plt_1 = ADRESH;

        // Reads a second time
        ADCON0 = 0b01001001; //AN0, sets GO bit
        GO = 1;
        while(GO ==1){}
        while(GO ==1){}
        char n_plt_2 = ADRESH;

        // take averages to get n_amb and n_plt
        //n_ambient = (n_amb_1+n_amb_2)/2;
        n_plt = (n_plt_1 + n_plt_2)/2;
        n_ambient = N_REF_AMB; //Hard coding ambient temp, sensor broken

        // calculates the bounds based on the ambient
        char delta_n_amb = n_ambient - N_REF_AMB;
        n_hot = N_REF_HOT+2*delta_n_amb;
        n_cold = N_REF_COLD+2*delta_n_amb;
        return;
}

void check_bounds(void)
{
        if(n_plt > n_hot)
        {
```

```
                PORTD = 0b00000010; // control fan
        }
        else if(n_plt < n_cold)
        {
                PORTD = 0b00000001;   // control heater
        }
        else
        {
        }
}

void    main (void)
{
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        init();


//&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
        while(1 != 2)
        {
                if(toggleswitch == 0)
                {
                        get_n_values();
                        check_bounds();
                }
                else
                {
                        PORTD = 0b00000011; // turn both devices off
                }

         }

//&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

}
```