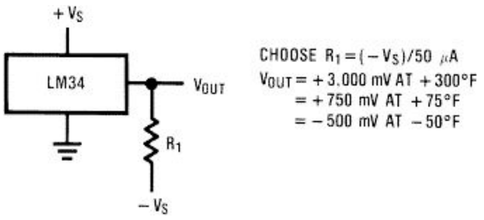
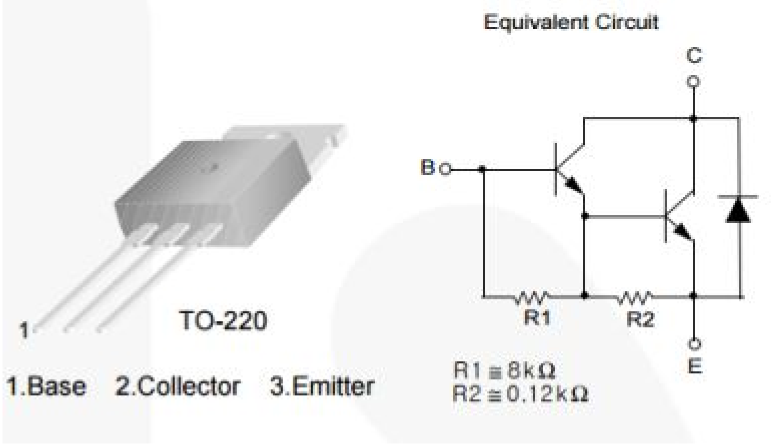
Objective: The purpose of this project was to create a thermostat. If the temperature of the room rose up the set temperature a fan would turn on in order to cool it back down. As with many other thermostats the set temperature point would vary depending on the time of day. Additionally the set temperature could be changed by turning the encoder knob in either direction.

Hardware: This project used an LM34 to convert temperature to voltage. This was used to read in the temperature of the room. Below is a schematic giving its details. The Vout was connected to the A0 pin on the arduino.



A TIP120 (Darlington Transistor) was also used to control the fan by turning it on or off. Below is the schematic for the TIP120. The base was connected to pin 12 of the arduino through resistors. The collector was connected to the fan and the emitter was connected to the ground.



Software: The three distinct software portions are for the clock, encoder knob, and temperature display. The code for the clock and encoder button will be displayed in Appendixes A), B), and C).

The loop code is displayed below as pseudo code.

```
If 1 second has passed
    Update and display clock on LCD
    Read Analog pin 0.
    Calculate room temperature using data from A0.
    If room temperature is above set temperature
```

```
        Turn fan on
    If room temperature is below set temperature
        Turn fan off
    While the current encoder position - set encoder position is > 4
        Increase set temperature
        Reset current encoder position
    While the current encoder position -set encoder position is < 4
        Decrease set temperature
        Reset current encoder position
    Print room temperature to LCD
    Print Set temperature to LCD
    Reset Timer
```

Testing: For testing I first changed the set temperature value using both the encoder knob and changing the time to ensure that fan would turn on and off. Next I held my fingers to the temperature sensor to ensure that the temperature readout would change. Without a control thermometer I based the displayed room temperature on personal assumption.

Appendix A) Code to update the clock

```
void UpdateClock()  //updates clock's values every second
{
 if(Seconds < 59)
 {
  Seconds++;  //increments seconds
 }
 else
 {
  Seconds = 0;  //resets seconds every 59 seconds
  if(Minutes<59)
  {
   Minutes++;  //increments Minutes
  }
  else
  {
   Minutes=0;  //resets minutes every 59 minutes
   if(Hours<23)
   {
    Hours++;  //increments hours
   }
   else
   {
     Hours = 0; //resets hours every 24 hours
     //reset
   }
  }
 }
}//end of UpdateClock
```

Appendix B) Code to send clock to LCD

```
void SendClock()
{
 LcdDriver.setCursor(0,0); //sets cursor position
 if(Hours<10)
 {
  LcdDriver.print("0"); //prints 0 if hours is <10
 }
 LcdDriver.print(Hours);//prints hours
 LcdDriver.print(":");
 if(Minutes<10)
 {
  LcdDriver.print("0");
 }
 LcdDriver.print(Minutes); //prints minutes
 LcdDriver.print(":");
 if(Seconds<10)
 {
  LcdDriver.print("0");
 }
 LcdDriver.print(Seconds); //prints seconds
}
```

Appendix C) Code to read encoder position

```
void monitorA()
{
 if(digitalRead(inputA)==digitalRead(inputB))  //reads input from encoder button
 {
   currEncPos ++;  //increment current encoder position
 }
 else
 {
   currEncPos --;  //decrement current encoder position
 }
}
//material from lab 6
void monitorB()//reads input from encoder button
{
 if(digitalRead(inputA) == digitalRead(inputB))
 {
   currEncPos--;  //decrement current encoder position
 }
 else
 {
   currEncPos++;  //increment current encoder position
 }
}
```

Appendix D) Code for setup and loop

```
void setup()
{
 LcdDriver.begin(16,2);
 LcdDriver.setCursor(0,0);
 LcdDriver.clear();

 attachInterrupt(0,monitorA,CHANGE);   //used to read encoder button
 attachInterrupt(1,monitorB,CHANGE);   //used to read encoder button

Serial.begin(9600);
 pinMode(fanPin, OUTPUT);  //sets fanPin (digital pin 12) as an output
```

```
  Hours = 7;  //initializes Hours at 7am. Must be changed to test hours function
  UseTemp();  //used to call method to initialize set temperature.
}

void loop()
{
  if(millis()-Timer>=Interval) //if 1 second has passed
  {
    UpdateClock(); //changes clock time
    SendClock(); //display clock time

    float fahr = (analogRead(A0))*(5.0/1023)*100;  //takes in voltage from LM34 and converts to degrees farenheit
    if(fahr>=setTemp)    //if room temperature is greater than set temperature
    {
     digitalWrite(fanPin, HIGH); //turn fan on
    }
    else if (fahr<setTemp) //if room temperature is less than set temperature
    {
     digitalWrite(fanPin, LOW); //turn fan off
    }


    while (currEncPos - encPos >4) //while encoder turns right
    {
     setTemp++;    //increase set temperature
     currEncPos = 0; //reset current encoder position
    }
    while (currEncPos - encPos <-4)    //while encoder turns left
    {
     setTemp--;  //decrease set temperature
     currEncPos = 0;   //reset current encoder position
    }


    LcdDriver.print("Room ");  //display room temperature in upper right hand corner (screen is too small to display full length)
    LcdDriver.print(fahr);
    LcdDriver.setCursor(0,1);  //change place of printing
    LcdDriver.print("Set Temp: "); //displays set temperature in lower left hand corner.
    LcdDriver.print(setTemp);
  }
}
```