#include <SoftwareSerial.h>
define rx 2
define tx 3

// define a string to hold incoming data from the PC
// define what pin rx is going to be
// define what pin tx is going to be
// define a string to hold the data from the Atlas Scientific product
// have we received all the data from the PC
// have we received all the data from the Atlas Scientific product
// used to hold a floating point number that is the RTD temperature

String inputstring = "";
String sensorstring = "";
boolean input_string_complete = false;
boolean sensor_string_complete = false;
float temperature;

// set up the hardware
// set baud rate for the hardware serial port_0 to 9600
// set baud rate for the software serial port to 9600
// set aside some bytes for receiving data from the PC
// set aside some bytes for receiving data from Atlas Scientific product

void setup() {
  Serial.begin(9600);
  myserial.begin(9600);
  inputstring.reserve(10);
  sensorstring.reserve(30);
}

// if the hardware serial port_0 receives a char
// read the string until we see a <CR>
// set the flag used to tell if we have received a completed string from the PC
void serialEvent() {
  inputstring = Serial.readStringUntil(13);
  input_string_complete = true;
}

// here we go...
// if a string from the PC has been received in its entirety
// send that string to the Atlas Scientific product
// add a <CR> to the end of the string
// clear the string
// reset the flag used to tell if we have received a completed string from the PC

void loop() {
  if (input_string_complete) {
    myserial.print(inputstring);
    myserial.print(‘\r’);
    inputstring = "";
    input_string_complete = false;
  }
  if (myserial.available() > 0) {
    char inchar = (char)myserial.read();
    sensorstring += inchar;
    if (inchar == ‘\r’)
      sensor_string_complete = true;
  }
  if (sensor_string_complete == true) {
    Serial.println(sensorstring);
    if (isdigit(sensorstring[0])) {
      temperature = sensorstring.toFloat();
      if (temperature >= 25.0)
        Serial.println("high");
      if (temperature <= 24.999)
        Serial.println("low");
    }
    sensorstring = "";
    sensor_string_complete = false;
  }
}

This code was written to be easy to understand.
Code efficiency was not considered.
Modify this code as you see fit.
This code will output data to the Arduino serial monitor.
For a description of the functions and monitor to control the RTD temperature circuit.
This code was written in the Arduino 1.6.7 IDE
An Arduino UNO was used to test this code.