## Design of a Temperature Controller with LCD Display using TI MSP430G2553

File: , , Alan Hutchenreuther

### Description

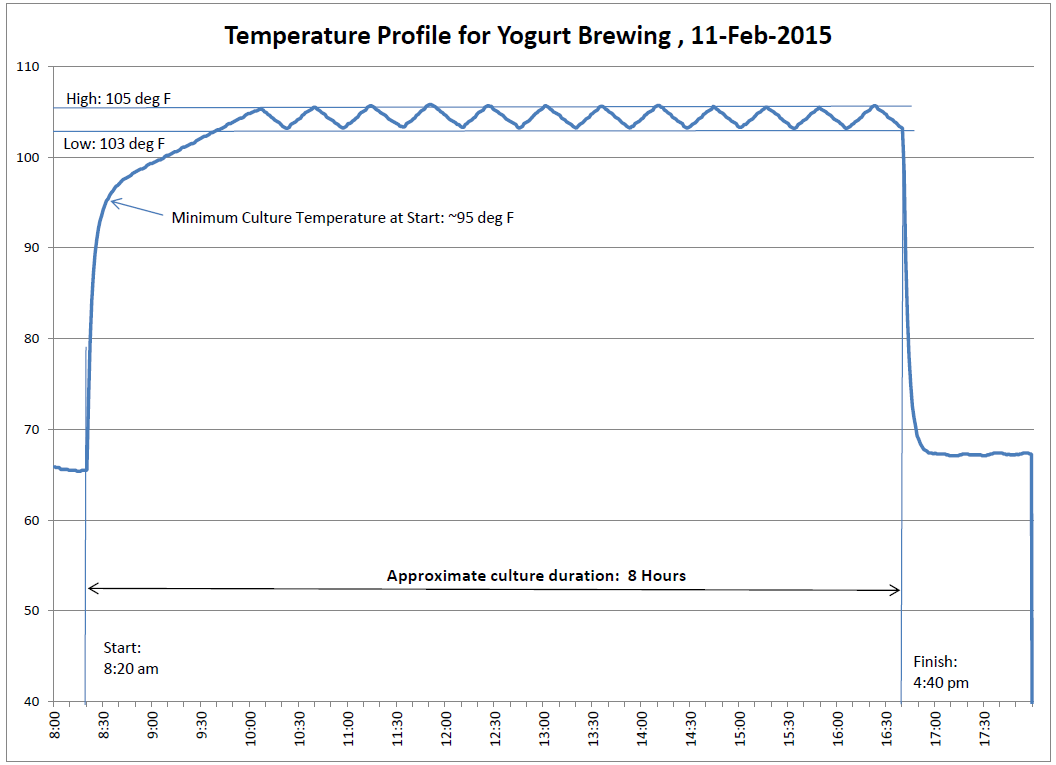
This describes construction of a temperature controlled box for culturing yogurt. Commercially available yogurt makers, such as the Salton maker, made 5 individual cups of yogurt for lunches. These are getting difficult to find. My approach is to make up to 1/2 gallon of yogurt in bulk. The yogurt recipe is found in the References. Yogurt making is not difficult aside from observing cleanliness and temperature control.

Temperature control within +/- 5 degrees F seems adequate, and this controller maintains a consistent temperature of 105 +/- 2 deg F. This controller is designed to simply turn a 40 watt incandescent appliance bulb ON and OFF inside the enclosure that holds the culture. The numeric display indicates the current box temperature, heater (lamp) % duty cycle, ON/OFF cycle count, minimum and maximum temperatures measured, respectively. See image below.

Other features:

1. The controller is programmed to switch heat ON and OFF +/- two degrees around the temperature set-point (hysteresis).
2. Minimum temperature is reset the first time the current temperature reaches the set point temperature. Otherwise minimum temperature will always show the room temperature at start-up... not very informative.
3. Safety features:
   1. Power switch has three-wire NEMA plug and receptacle, 120Vac with ground.
   2. Controls are opto-isolated from power switch.
   3. If the measured temperature falls outside the tolerance band of 10 to 140 deg F, the heating control disables heating, and the display will indicate ERROR. This could happen if the temperature sensor or its connections fail.

Yogurt can be made without any fancy controller, but I did this as a learning exercise for Arduino and TI MSP-430 microcontroller applications. I think the consistent temperature does give a more predictable result in all seasons of the year.



### 

Figure - Controller in enclosure, powered up at end of Yogurt culture run.

### C:\Users\Alan\Documents\Arduino\Project Documentation\Yogurt Culture Controller\11-8-2014 Yogurt Culture Controller\IMG_5732.JPG

Figure - Microcontroller and Power Switch Tail. Sensor is on the left, laying on white sheet.

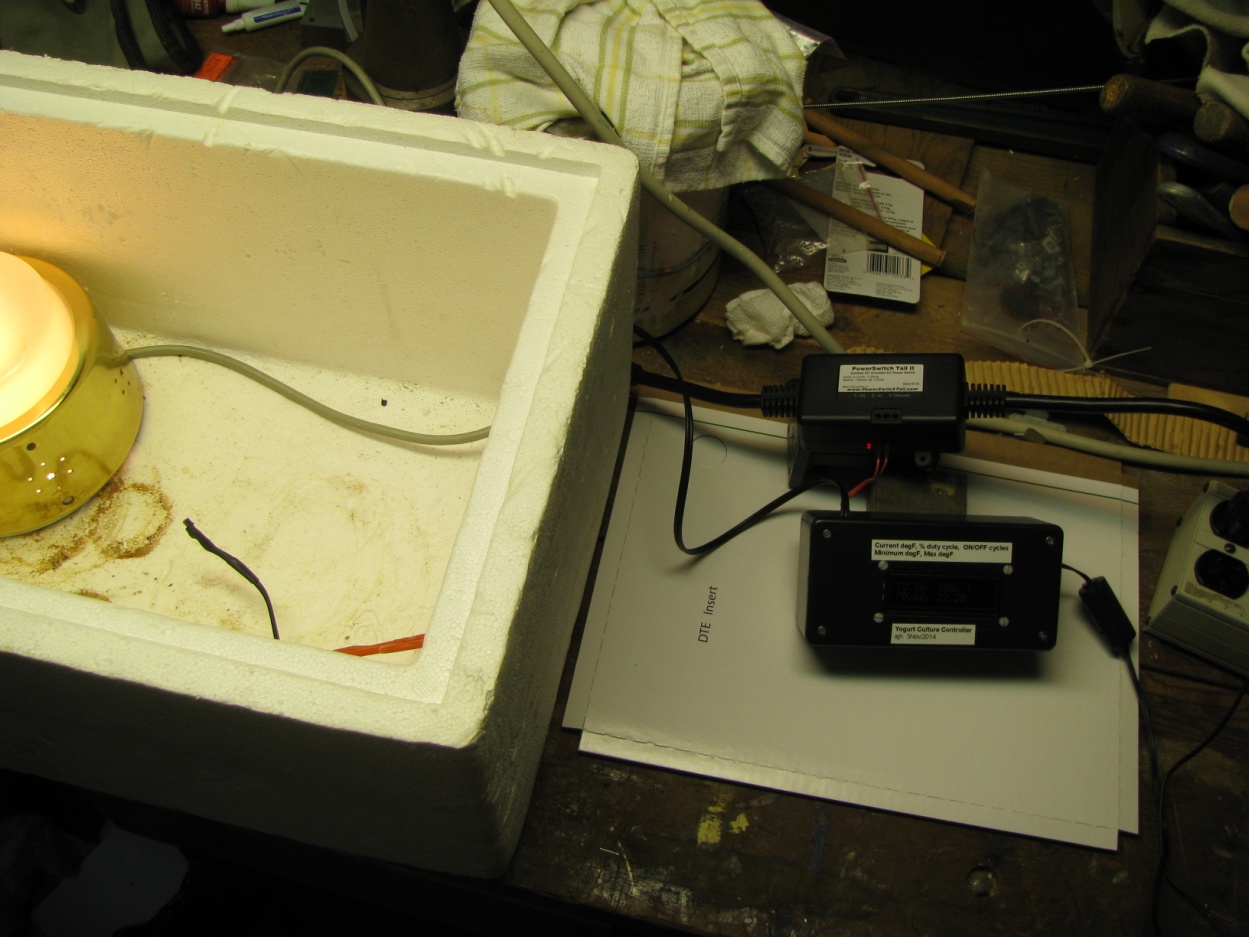


Figure - Put heat source (lamp) and sensor in insulated box with cover. Ready for operation.

### References:

1. <MyHomemadeYogurt.pdf>, Recipe for making yogurt.
2. MSP430 Family Users Guide. [MSP430x2xx Family Users Guide slau144j.pdf](MSP430x2xx%20Family%20Users%20Guide%20slau144j.pdf)
3. MSP430G2553 Datasheet. <msp430g2553.pdf>
4. <ADM1602K-NSA-FBS-3.3v.pdf>, LCD Display datasheet.
5. [st7066 LCD Controller for ADM1602K.pdf](st7066%20LCD%20Controller%20for%20ADM1602K.pdf), ST7066 LCD controller on ADM1602K
6. [ADAfruit Tutorial character-lcds.pdf](ADAfruit%20Tutorial%20character-lcds.pdf), General application information on LCD character displays.
7. <TMP35_36_37.pdf>, Low Voltage Temperature Sensors TMP-35/36/37

### Key Components:

1. TI MSP-EXP430F5529LP LaunchPad (Replaces a more expensive Arduino Uno R3)
2. *43oh - MSP430 Launchpad Prototyping PCB*, DEV\_LPB-PROTO\_PCB (Formerly LaunchPadBling).
3. Basic Parallel LCD display. SparkFun ADM1602K 3.3V Amber on Black.
4. Temperature Sensor - TMP36GZ voltage output.
5. Enclosure- Radio Shack (6" x 2.75" x 2" approx.)
6. 5Vdc adapter with mini-USB plug.
7. PowerSwitch Tail II

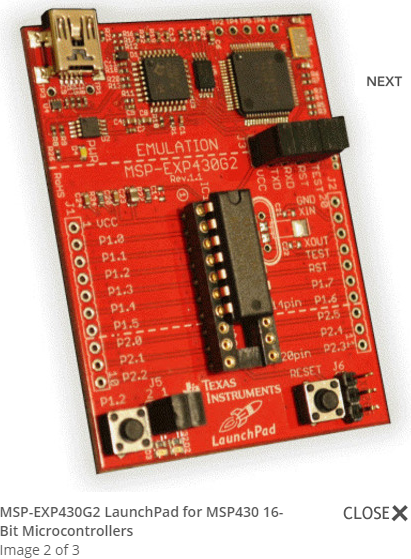


Figure MSP-EXP LaunchPad Version 1.4 sample

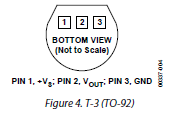


Figure - Temperature Sensor with analog voltage output.

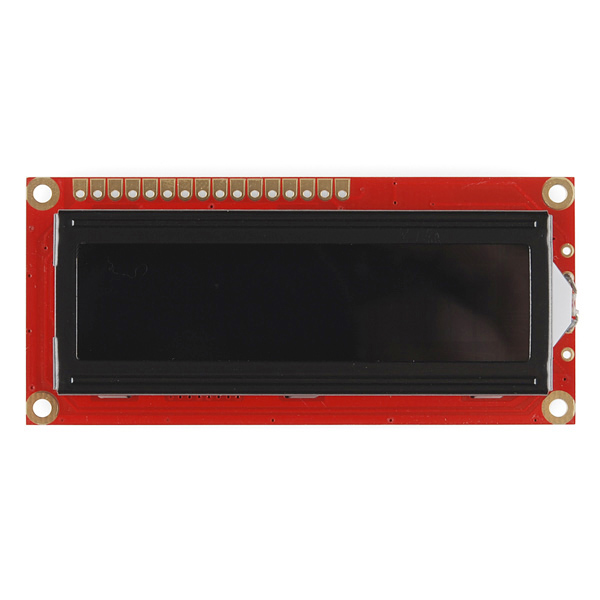
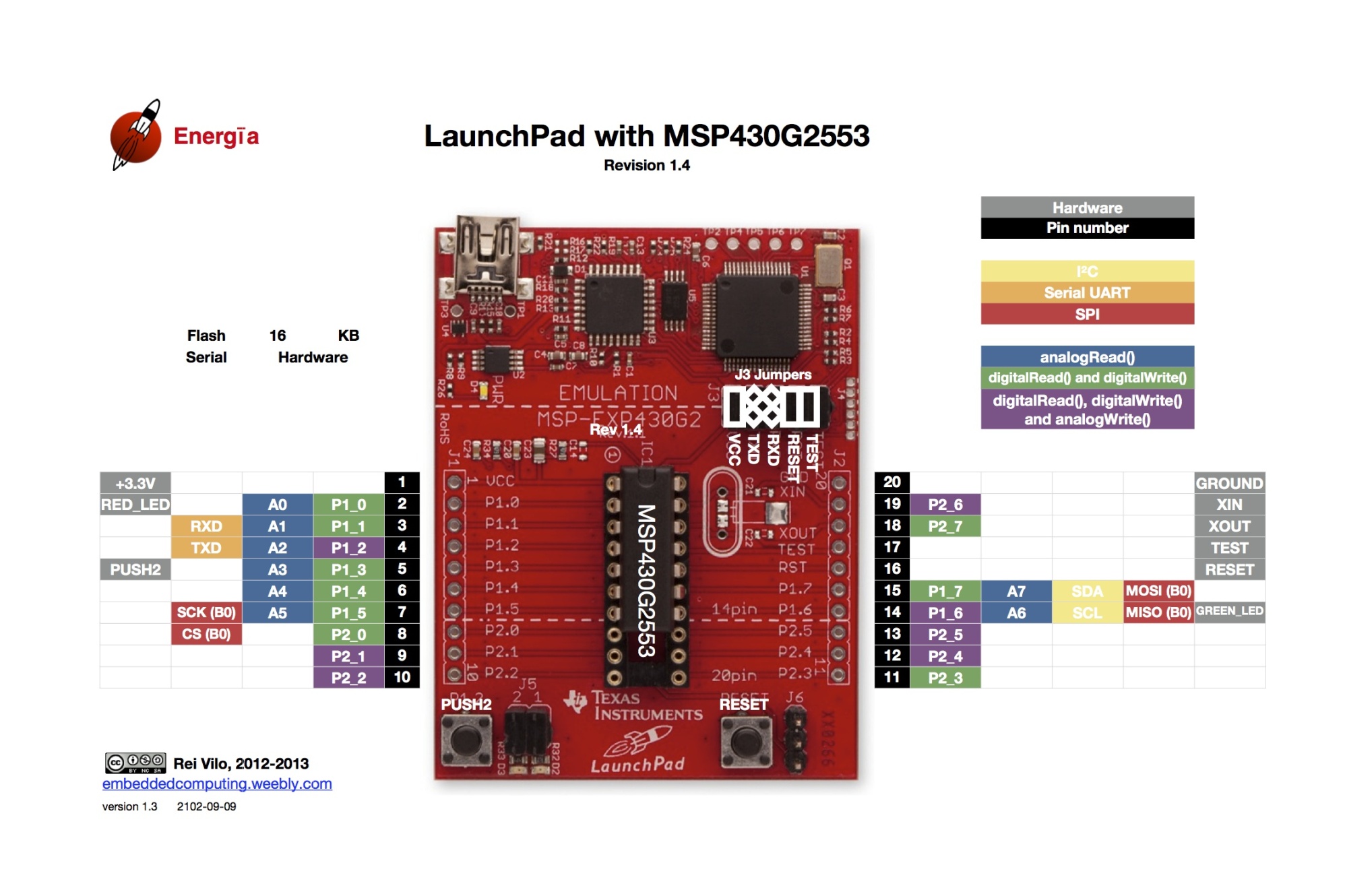


Figure - LCD Display (pin 1 on left)



Figure - Power Switch Tail II

### Wiring Pin Assignments.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MSP430G2553 Signal** | **LaunchPad Pin** | **External Device** | **Device Signal** | **Device Pin** |
| Gnd | 20 | LCD display | Vss | 1 |
| Vcc | 1 | LCD display | Vcc | 2 |
|  |  | LCD display | Vo (contrast pot wiper) | 3 |
| P2\_0 | 8 | LCD display | RS (register select) | 4 |
| Gnd | 20 | LCD display | R/W (read/write) | 5 |
| P2\_1 | 9 | LCD display | Enable | 6 |
| P2\_2 | 10 | LCD display | D4 | 11 |
| P2\_3 | 11 | LCD display | D5 | 12 |
| P2\_4 | 12 | LCD display | D6 | 13 |
| P2\_5 | 13 | LCD display | D7 | 14 |
| Vcc | 1 | LCD display | Backlight LED+ | 15 |
| Gnd | 20 | LCD display | Backlight LED- | 16 |
| Vcc | 1 | LED Pot | Pot+ |  |
| Gnd | 20 | LED Pot | Pot- |  |
| P1\_7 | 15 | PowerSwitch Tail II | (ON/OFF) (Orange lead) | + |
| Gnd | 20 | PowerSwitch Tail II | (Return) (Red lead) | - |
| Vcc | 1 | Temperature Sensor | +Vd | 1 \* |
| P1\_0 | 2 | Temperature Sensor | Vout | 2 |
| Gnd | 20 | Temperature Sensor | GND | 3 \* |
| Red\_LED | 2 | Jumper OFF | Launchpad J5-2 | N/C |
| Green\_LED | 14 | Jumper ON | Launchpad J5-1 | N/C |

\* *Add 100nF capacitor across temperature sensor power leads to reduce noise on Vo readings.*

### Development Environment:

Energia (energia-0101E0013)

### Software Libraries

LiquidCrystal   
(C:\Users\Alan\Documents\Energia\energia-0101E0013\hardware\msp430\libraries\LiquidCrystal)

### Sketch Location:

### C:\Users\Alan\Documents\Energia\TemperatureControl\_MSP430\_20Oct\TemperatureControl\_MSP430\_20Oct.ino

### 

### Appendix: Sample code listing.

This code is current at the time of writing this description. Any modifications since this time will be found in the source code file.

/\*

This is a conversion of an Arduino application to run on a Launchpad v1.4 with MSP430G2553.

Development environment is Energia 0101E0012 or later.

Temperature measurement adapted from SD card datalogger.

created 24 Nov 2010

modified 9 Apr 2012

by Tom Igoe

This example code is part of the public domain

The purpose of this code is to control the yogurt brewing temperature.

This is done by maintaining a 105 degF chamber temperature through ON/OFF control of an incandescent lamp.

A histeresis band is set about the setpoint temperature to reduce the

number of heating/cooling cycles and preserve relay life.

Safety feature:

Prevent heater from being stuck ON if temperature sensor or connection fails.

Revision History

12Sept14

G2231 V1.4 does not fit.

G2553 V1.4 4,495 bytes (of a 16,384 byte max) (Float library, no LCD, Heater control)

G2553 V1.4, 7,631 bytes (of a 16,384 byte max) (Float, no LCD, Heater control, Software Serial library)

G2553 V1.4, 7,670 bytes (of a 16,384 byte max) (Float, no LCD, Heater control, Software Serial library, GREEN\_LED blink)

18Oct14

G2553 V1.4 4532 bytes (of a 16,384 byte max) (Float, Heater control, GREEN\_LED blink, remove Software Serial lib. Add LiquidCrystal.lib

Issues: Serial does not work. Forum shows corrective actions that have not worked for me yet.

Issue working on: re-define pins that drive display.

20Oct14

G2553 V1.4 7,677 bytes (of a 16,384 byte maximum) (Float, Heater control, GREEN\_LED blink, remove Software Serial lib. Add LiquidCrystal.lib)

LCD display pins redefined & wired on adapter board.

HEATER\_PIN definition changed from 7 to P1\_7 for G2553

ajh141023 code version 4.1 - Change analog reference from Default to 1.5 V. Upper temperature measurement limit of ADC is 212 degF

\*/

#include <LiquidCrystal.h>

#define TEMP\_PIN A0

#define HEATER\_PIN P1\_7

#define MEASUREMENTS\_TO\_SUM 10

#define ANALOG\_REF 1.5 // ajh141023

#define DELAY\_MS 100

#define TSP 105 // degF

#define TTOL 2

#define HIGH\_T\_LIMIT 140.0 // degF

#define LOW\_T\_LIMIT 10.0

// lcd arguments: RS, EN, DB4, DB5, DB6, DB7 - signals on LCD pins 4, 6, 11, 12, 13, 14 respectively.

LiquidCrystal lcd(P2\_0, P2\_1, P2\_2, P2\_3, P2\_4, P2\_5); //ajh20141020 - Rewire LCD to free up P1\_0 (A0, TEMP\_PIN), P1\_6 (GREEN\_LED), P1\_7 (HEATER\_PIN)

float tempK, tempC, tempF, temp;

float thisTempF = 0.0;

float sumT = 0.0;

long count = 0;

float maxT = -10.0;

float minT = 150.0;

float timeSinceStart = 0.0;

float heaterONtime = 0.0;

boolean TSPreached = false;

long cycleCount= 0;

boolean heaterOFF;

boolean errorFlag = false;

float heaterDutyCycle = 0.0;

boolean loopstate = true; // Green LED control.

boolean debugging = false;

void setup() {

analogReference( INTERNAL1V5); // ajh141023

// set up the sensor pin as an input

pinMode(TEMP\_PIN, INPUT);

// initialize control pin as output

pinMode(HEATER\_PIN, OUTPUT);

pinMode(GREEN\_LED, OUTPUT);

digitalWrite( HEATER\_PIN, LOW);

heaterOFF = true;

// Liquid Crystal //ajh20141019

lcd.begin(16,2); // using 16 x 2 LCD

lcd.setCursor( 0, 0);

lcd.print("Temperature");

lcd.setCursor( 1, 1);

lcd.print("Controller v4.1"); // Update version number here!

}

void loop() {

// This loop inputs sensorVal every DELAY\_MS ms

// thisTempF contains the average of the previous 10 measurements in degF.

float voltage;

int sensorVal;

float elapsedMinutes;

// Read temperature sensor

sensorVal = analogRead(TEMP\_PIN);

voltage = sensorVal \* ANALOG\_REF/1024.0;

tempC = (voltage - 0.5) \* 100.0; // TMP36 Sensor has 0.5V offset to allow negative T readings on unipolar supply.

tempF = ((tempC \* 9.0)/5.0) + 32.0;

sumT += tempF;

elapsedMinutes = (1.0 \* count \* DELAY\_MS)/60000;

count++;

if (count >= MEASUREMENTS\_TO\_SUM ) {

// Calculate current average reading.

thisTempF = sumT/MEASUREMENTS\_TO\_SUM;

count = 0;

sumT = 0.0;

// Detect sensor failure and disable heater.

if( (thisTempF >= HIGH\_T\_LIMIT) || (thisTempF <= LOW\_T\_LIMIT)) {

errorFlag = true;

heaterOFF = true;

digitalWrite( HEATER\_PIN, LOW);

}

// Record Max and Min T for display purposes

timeSinceStart += elapsedMinutes;

if (thisTempF > maxT) {

maxT = thisTempF;

}

else if (thisTempF < minT) {

minT = thisTempF;

}

// Update heater control.

if( !errorFlag){

if (thisTempF < (TSP - TTOL)) {

// Turn heater ON

digitalWrite( HEATER\_PIN, HIGH);

// Statistics

if( heaterOFF) {

cycleCount++;

heaterOFF = false;

}

}

else if ( thisTempF > (TSP + TTOL)) {

// Turn heater OFF

digitalWrite( HEATER\_PIN, LOW);

heaterOFF = true;

}

} // !ErrorFlag

// Statistical analysis

// When Temperature setpoint is reached, reset the min Temp to current temp.

if( !TSPreached & (thisTempF >= TSP)) {

TSPreached = true;

minT = thisTempF;

};

// Update duty cycle

if( !heaterOFF) {

heaterONtime += elapsedMinutes;

}

heaterDutyCycle = 100.0 \* (heaterONtime / timeSinceStart);

// Blink GREEN\_LED

digitalWrite( GREEN\_LED, loopstate);

loopstate = !loopstate;

// Update LCD

lcd.clear();

lcd.setCursor( 0, 0);

// First line

lcd.print( thisTempF); lcd.print( ", ");

if ( !errorFlag) {

if( debugging) {

lcd.print( voltage \* 1000);

}

else {

lcd.print( int(heaterDutyCycle)); lcd.print( ", ");

lcd.print( cycleCount);

}

}

else {

lcd.print( " Error");

}

// Second line

lcd.setCursor( 0, 1);

if( debugging) { // ajh141023 - for determination of reference voltage.

lcd.print( sensorVal);

}

else {

lcd.print( minT); lcd.print( ", ");

lcd.print( maxT);

}

}; // count >= MEASUREMENTS\_TO\_SUM

delay( DELAY\_MS);

}// loop.