Sarah Wood

Embedded Programming

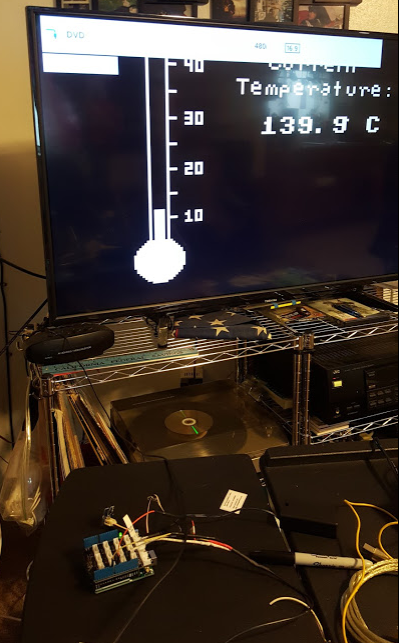
Chapter 8

Assignment 1

Build custom RCA cable



Demonstrate running Thermometer on TV



Still needs a little tuning between the sensor and the output. But it works!

Include Source code

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

ThermometerTV.cpp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#include <TVout.h>

#include <fontALL.h>

#include "thermometer.h"

const float SUPPLY\_VOLTAGE = 5.0;

const float MIN\_TEMP = 5.5;

const float MAX\_TEMP = 40.0;

const unsigned int SCREEN\_WIDTH = 120;

const unsigned int SCREEN\_HEIGHT = 96;

const unsigned int TEMP\_SENSOR\_PIN = A0;

const unsigned int SCALE\_X\_MIN = 8;

const unsigned int SCALE\_Y\_MIN = 6;

const unsigned int SCALE\_Y\_MAX = 75;

const unsigned int SCALE\_WIDTH = 3;

const unsigned int SCALE\_HEIGHT = SCALE\_Y\_MAX - SCALE\_Y\_MIN;

float current\_temperature = 0.0;

unsigned long last\_measurement = millis();

TVout TV;

void setup() {

TV.begin(NTSC, SCREEN\_WIDTH, SCREEN\_HEIGHT);

TV.bitmap(0, 1, thermometer);

TV.select\_font(font4x6);

TV.set\_cursor(20, 4);

TV.print("40");

TV.set\_cursor(20, 24);

TV.print("30");

TV.set\_cursor(20, 44);

TV.print("20");

TV.set\_cursor(20, 64);

TV.print("10");

}

void loop() {

unsigned long current\_millis = millis();

if (abs(current\_millis - last\_measurement) >= 1000) {

current\_temperature = get\_temperature();

last\_measurement = current\_millis;

int y\_pos = mapfloat(

current\_temperature, MIN\_TEMP, MAX\_TEMP, SCALE\_Y\_MAX, SCALE\_Y\_MIN);

TV.draw\_rect(

SCALE\_X\_MIN, SCALE\_Y\_MIN, SCALE\_WIDTH, SCALE\_HEIGHT, BLACK, BLACK);

TV.draw\_rect(

SCALE\_X\_MIN, y\_pos, SCALE\_WIDTH, SCALE\_Y\_MAX - y\_pos, WHITE, WHITE);

TV.select\_font(font6x8);

TV.set\_cursor(53, 1);

TV.print("Current");

TV.set\_cursor(40, 11);

TV.print("Temperature:");

TV.select\_font(font8x8);

TV.set\_cursor(50, 25);

TV.print(current\_temperature, 1);

TV.print(" C");

TV.draw\_circle(88, 27, 1, WHITE);

}

}

const float mapfloat(

float x, float in\_min, float in\_max, float out\_min, float out\_max)

{

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

const float get\_temperature() {

const int sensor\_voltage = analogRead(TEMP\_SENSOR\_PIN);

const float voltage = sensor\_voltage \* SUPPLY\_VOLTAGE / 1024;

return (voltage \* 1000 - 500) / 10;

}

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

thermometer.cpp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#include <Arduino.h>

#include <avr/pgmspace.h>

#include "thermometer.h"

PROGMEM const unsigned char thermometer[] = {

20, 94,

B00000000, B11110000, B00000000,

B00000001, B00001000, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 40.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

// ...

B00000010, B00000111, B10000000, // 35.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 30.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 25.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 20.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 15.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000111, B10000000, // 10.0

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000,

B00000010, B00000100, B00000000, // 5.5

B00000111, B11111110, B00000000,

B00001111, B11111111, B00000000,

B00011111, B11111111, B10000000,

B00111111, B11111111, B11000000,

B01111111, B11111111, B11100000,

B01111111, B11111111, B11100000,

B11111111, B11111111, B11110000,

B11111111, B11111111, B11110000,

B11111111, B11111111, B11110000,

B11111111, B11111111, B11110000,

B11111111, B11111111, B11110000,

B11111111, B11111111, B11110000,

B01111111, B11111111, B11100000,

B01111111, B11111111, B11100000,

B00111111, B11111111, B11000000,

B00011111, B11111111, B10000000,

B00001111, B11111111, B00000000,

B00000111, B11111110, B00000000,

B00000001, B11111000, B00000000,

B00000001, B11111000, B00000000,

};

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

thermometer.h

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/\*\*\*

\* Excerpted from "Arduino: A Quick-Start Guide, Second Edition",

\* published by The Pragmatic Bookshelf.

\* Copyrights apply to this code. It may not be used to create training material,

\* courses, books, articles, and the like. Contact us if you are in doubt.

\* We make no guarantees that this code is fit for any purpose.

\* Visit http://www.pragmaticprogrammer.com/titles/msard2 for more book information.

\*\*\*/

#ifndef THERMOMETER\_H

#define THERMOMETER\_H

extern const unsigned char thermometer[];

#endif