Sarah Wood

Embedded Programming

Chapter 13

Assignment 2

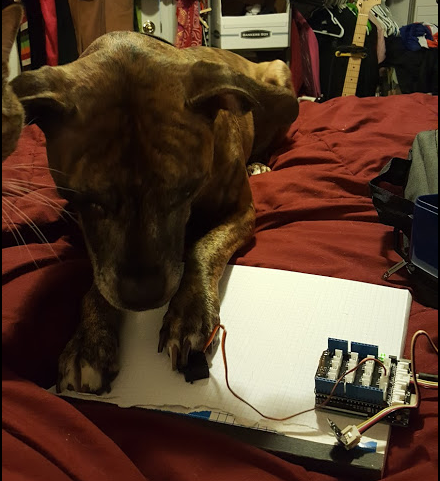
Create a thermometer based on a TMP36 temperature sensor and a servo

motor. Its display could look like the image below; that is, you have to

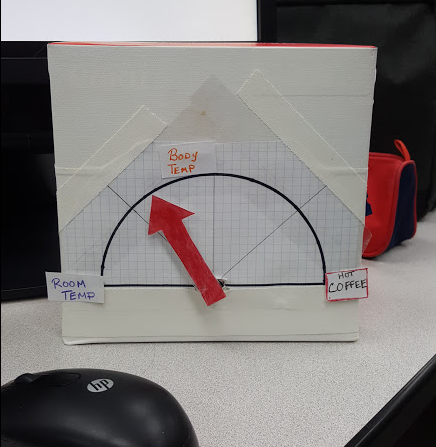
move an arrow that points to the current temperature.



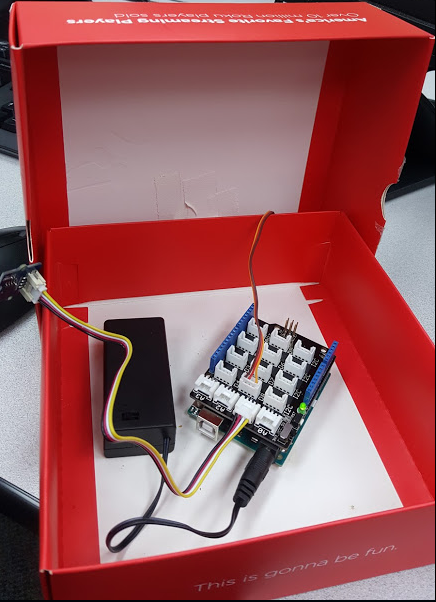
Screen Caps



Testing servo calibration at home. It fascinated the dog and cat. I could have some fun building pet toys as a side project.



It’s cooling down to room temp after I held the sensor in my hand for a minute. I set the scale to actually show some movement. (Most thermometers just sit there: what fun is that?)



The view from backstage. I can close the box and everything still works. I’d need to put in a punch-out for the sensor if I wanted it to be actually accurate.

Source Code

(the TMP36 headers are the same as ever.)

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

Source.cpp

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

#include <Servo.h>

#include "thermometer.h"

const unsigned int MAX\_MEMBERS = 10;

const unsigned int TEMP\_SENSOR\_PIN = A1;

const float SUPPLY\_VOLTAGE = 5.0; //tmp36 supply

float current\_temperature = 0.0;

unsigned long last\_measurement = millis();

#include <Servo.h>

const unsigned int MOTOR\_PIN = 8;

const unsigned int MOTOR\_DELAY = 15;

const unsigned int BAUD\_RATE = 9600;

const unsigned int SERIAL\_DELAY = 5;

Servo blaminatr;

void setup() {

Serial.begin(BAUD\_RATE);

blaminatr.attach(MOTOR\_PIN);

blaminatr.write(160);

}

void loop() {

unsigned long current\_millis = millis();

current\_temperature = get\_temperature();

last\_measurement = current\_millis;

//temp indicated = max-measured

//~180 degrees movmt, ~40 deg temp change (0-40c)

//we will cheat and only use 160 degrees of movement

//unsigned long pointerDegrees = (160-(current\_temperature\*4));

unsigned long pointerDegrees = (160-((current\_temperature-18)\*15)); //max movement for testing at room temps

Serial.print(current\_temperature);

Serial.println (" deg C");

blaminatr.write(pointerDegrees);

delay(5000);

/\*

char name[MAX\_NAME + 1];

if (Serial.available()) {

unsigned int i = 0;

while (Serial.available() && i < MAX\_NAME + 1) {

const char c = Serial.read();

if (c != -1 && c != '\n')

name[i++] = c;

delay(SERIAL\_DELAY);

}

name[i] = 0;

Serial.print(name);

Serial.println(" is to blame.");

blaminatr.blame(name);

}

\*/

}

const float get\_temperature() {

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

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

return (voltage \* 1000 - 500) / 100;

}