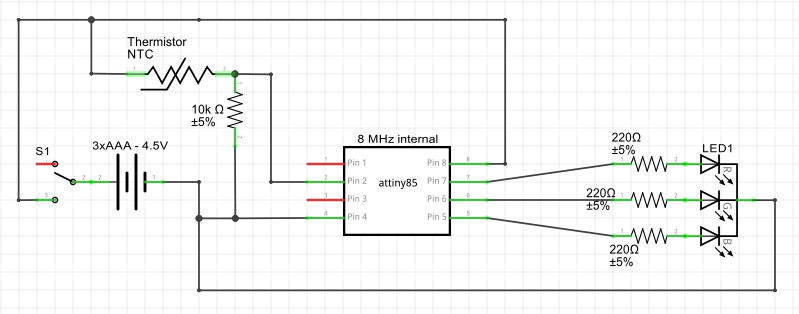
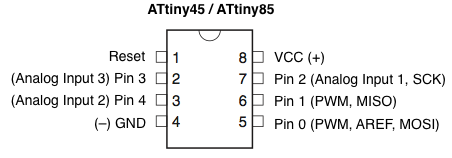
RGB LED Faucet – AtTiny85



Self-motivated project designed to change water color according to temperature of water. Uses a 10kΩ thermistor with the Steinhart Equation. Includes a switch to operate and 3xAAA batteries for power. Color ranges from Dark Blue to Aqua, Green, Yellow, Orange, and Red, and is designed to flash red if water is dangerously hot (115°F or greater)

Issues:

* The AtTiny85 only has two PWM pins, so software PWM was needed on analog input pin 2
* Original temperature scale ranged from 32-120°F, however town water only gets as cold as 60°F so scale was revised



Scale:

**Color** **RGB – (xxx-xxx-xxx) from 0 to 255**

< 70°F – **Dark Blue 000-000-255**

70 to 79°F – **Aqua 000-255-255**

79 to 88°F – **Green 000-255-000**

88 to 97°F – **Yellow 255-100-000**

97 to 106°F – **Orange 255-030-000**

106 to 115°F – **Red 255-000-000**

> 115°F – **Flashing Red 255-000-000**

Code:

const int red = 2;

const int green = 1;

const int blue = 0;

const int sensor = 3;

const int del = 75; //minimum time for changes to take place

void setup() {

pinMode(red, OUTPUT);

pinMode(green, OUTPUT);

pinMode(blue, OUTPUT);

pinMode(sensor, INPUT);

}

void loop() {

double temperature;

temperature = getTemp(analogRead(sensor));

if (temperature <= 70) {

makeBlue();

}

else if (temperature > 70 && temperature <= 79) {

makeAqua();

}

else if (temperature > 79 && temperature <= 88) {

makeGreen();

}

else if (temperature > 88 && temperature <= 97) {

makeYellow();

}

else if (temperature > 97 && temperature <= 106) {

makeOrange();

}

else if (temperature > 106 && temperature <= 115) {

makeRed();

}

else {

flashRed(100);

}

}

double getTemp(int RawADC) {

double Temp;

Temp = log(10000.0\*((1024.0/RawADC-1)));

Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 \* Temp \* Temp ))\* Temp );

Temp = Temp - 273.15;

Temp = (Temp \* 9.0)/ 5.0 + 32.0;

return Temp;

}

void makeRed()

{

digitalWrite(red, HIGH);

digitalWrite(green, LOW);

digitalWrite(blue, LOW);

delay(del);

}

void makeGreen()

{

digitalWrite(red, LOW);

digitalWrite(green, HIGH);

digitalWrite(blue, LOW);

delay(del);

}

void makeBlue()

{

digitalWrite(red, LOW);

digitalWrite(green, LOW);

digitalWrite(blue, HIGH);

delay(del);

}

void makeYellow()

{

digitalWrite(red, HIGH);

digitalWrite(blue, LOW);

long start = millis();

long endtime = start;

while((endtime - start) <= del)

{

digitalWrite(green, HIGH);

delayMicroseconds(100);

digitalWrite(green, LOW);

delayMicroseconds(255);

endtime = millis();

}

}

void makeOrange()

{

digitalWrite(red, HIGH);

digitalWrite(blue, LOW);

long start = millis();

long endtime = start;

while((endtime - start) <= del)

{

digitalWrite(green, HIGH);

delayMicroseconds(30);

digitalWrite(green, LOW);

delayMicroseconds(255);

endtime = millis();

}

}

void makeAqua()

{

digitalWrite(red, LOW);

digitalWrite(green, HIGH);

digitalWrite(blue, HIGH);

delay(del);

}

void flashRed(int d)

{

digitalWrite(green, LOW);

digitalWrite(blue, LOW);

long start = millis();

long endtime = start;

while((endtime - start) <= del)

{

digitalWrite(red, HIGH);

delay(d);

digitalWrite(red, LOW);

delay(d);

endtime = millis();

}

}