ANEXO I. Código de programación de Arduino.

```
#include <SPI.h>

byte addresspot1 = B00010000;
byte addresspot0 = B00000000;
const float A = 1000;     //Darkness resistance in KΩ
const float B = 5;        //Light resistance in KΩ
const float Rc = 0.22;       //Calibration resistance in KΩ
int CS= 10, aux = 0;
const int analogPin = A5;
const int analogPin1 = A4;
const int LDRPin = A0;
const int NTCPin = A2;
float bright, voltage, current, current1, V;

void setup()
{
Serial.begin(9600);
pinMode (CS, OUTPUT);
Serial.println("CLEARSHEET");
Serial.println("LABEL,Date,Time,Timer,Voltage,Current,Power,Bright,Temp");
SPI.begin();
}

void loop()
{
  delay(6500);

  for (int i = 0; i <= 256; i++){
  int j= 256-i;
  digitalPot1(i);
  digitalPot0(j);
  aux++;
  if (aux%2 != 0){delay(10);}
  if (aux%2 == 0){
    for (int i = 0; i <= 2; i++){
      float cur = analogRead(analogPin1);
      float vol = analogRead(analogPin);
      V = analogRead(LDRPin);
      voltage =conversionvoltage(vol);
      current = conversioncurrent(cur);
      bright = brightness(V);
      float power = getpower(voltage,current);
     // current1 = conversionvoltage(cur);
     float temp[3];
     getTemp(temp);
      Serial.print( /*(String)*/ "DATA,DATE,TIME,TIMER,");
      Serial.print(voltage);
      Serial.print(",");
```

```
      Serial.print(current);
      Serial.print(",");
      Serial.print(power);
      Serial.print(",");
      Serial.print(bright);
      Serial.print(",");
      Serial.println(temp[1]);//Last row sending data MUST finish with Serial.println
      delay(300);}
    }
  if (aux >= 256){
    aux = 0;}
}
delay(2000);


for (int i = 256; i >= 0; i--){
  int j=256-i;
  digitalPot1(i);
  digitalPot0(j);
  aux++;                          //Incluir delay
  if (aux%2 != 0){delay(10);}
  if (aux%2 == 0){
    for (int i = 0; i <= 2; i++){
      float cur = analogRead(analogPin1);
      float vol = analogRead(analogPin);
      V = analogRead(LDRPin);
      voltage =conversionvoltage(vol);
      current = conversioncurrent(cur);
      bright = brightness(V);
      float power = getpower(voltage,current);
     // current1 = conversionvoltage(cur);
     float temp[3];
     getTemp(temp);
      Serial.print( /*(String)*/ "DATA,DATE,TIME,TIMER,");
      Serial.print(voltage);
      Serial.print(",");
      Serial.print(current);
      Serial.print(",");
      Serial.print(power);
      Serial.print(",");
      Serial.print(bright);
      Serial.print(",");
      Serial.println(temp[1]);//Last row sending data MUST finish with Serial.println
      delay(300);}
    }
  if (aux >= 256){
    aux = 0;}
}
delay(2000);

}
```

```
int digitalPot1(int valuepot1)
{
  digitalWrite(CS, LOW);
  SPI.transfer(addresspot1);
  SPI.transfer(valuepot1);
  digitalWrite(CS, HIGH);
}

int digitalPot0(int valuepot0)
{
  digitalWrite(CS, LOW);
  SPI.transfer(addresspot0);
  SPI.transfer(valuepot0);
  digitalWrite(CS, HIGH);
}

//IN CASE YOU WANT TO LEARN MORE ABOUT MCP4251: http://www.learningaboutelectronics.com/Articles/MCP4231-dual-digital-potentiometer-circuit.php

float conversionvoltage(float x){
  return (5*x/1023);
}

float conversioncurrent(float x){
  float volts = (5*x/1023);
  float aux = abs (volts-2.5);
  float currentmA = (10*aux/0.255);
  return (currentmA);
}

float getpower(float x, float y){
  return(x*y);
  }

float brightness(float ilum){   //IN CASE YOU WANT TO LEARN MORE ABOUT LDR: https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoresistencia-ldr/
  ilum = ((long)(1024-V)*A*10)/((long)B*Rc*V);  //use if LDR is between GND and A0
  // ilum = (V*A*10)/(B*Rc*(1023-V));    //use if LDR is between A0 and Vcc

  return (ilum);
  }

void getTemp(float * t){    //IN CASE YOU WANT TO LEARN MORE ABOUT NTC: https://www.jameco.com/Jameco/workshop/techtip/temperature-measurement-ntc-thermistors.html
  const int analogPin = A2;
  const float invBeta = 1.00 / 3997.00;

  const  float adcMax = 1023.00;
  const float invT0 = 1.00 / 298.15;   // room temp in Kelvin

  int i, numSamples = 5;
  float  K, C, F,  adcVal;
```

```
  adcVal = 0;
  for (i = 0; i < numSamples; i++)
   {
     adcVal = adcVal + analogRead(analogPin);
     delay(100);
   }
  adcVal = adcVal/5;
  K = 1.00 / (invT0 + invBeta*(log ( adcMax / (float) adcVal - 1.00)));
  C = K - 273.15;                    // convert to Celsius
  F = ((9.0*C)/5.00) + 32.00;   // convert to Fahrenheit
  t[0] = K; t[1] = C;
   // adcVal = adcVal*5/1023;
//  t[2] =   adcVal;
  return;
  }
```