

## ANNEX C – PROGRAMARI

A continuació hi ha el codi complet per a fer funcionar el llit refrigerat amb Arduino UNO, amb comentaris que expliquen què passa a cada línia i separen els diferents apartats, per a una millor comprensió.

```
/*Cold bed for the BCN3D+ printer
There are two main circuits: one is a temperature sensor using a thermistor and
the other one controls the power supplied to the Peltiers the sensor reads the
temperature at the center of the bed. If this temperature is more than 0.5°C lower
than the target temperature, it decreases the power supplied to the peltiers, and
viceversa. The target temperature, actual temperature, and power supplied are
shown on the serial monitor.
In addition, a led turns on when the temperature equals the target temperature +-
0.5°C.
*/

#include <math.h>           //loads the more advanced math functions

// Definition of the target temperature and tolerance interval
float TargetTemp = 17.0;   // Target temperature in Celcius
float Tolerance = 0.5;     // Tolerance interval in Celcius

//Calculate lower and upper temperature
double Tmin = (TargetTemp - Tolerance);
double Tmax = (TargetTemp + Tolerance);

// Constants (pin numbers):
int peltier = 3;           //The N-Channel MOSFET is on digital pin 3
const int ledPin = 13;    // the number of the LED pin

// Variables:
int ledState = LOW;       // ledState used to set the LED
double power = 50;        // Power level fro 0 to 99%
int peltier_level = map(power, 0, 99, 0, 255); //This is a value from 0 to 255
that actually controls the MOSFET
int centivolt = map(power, 0, 99, 0, 500); //Value from 0 to 500 showing the
centivolts sent to the Peltier
double voltage = centivolt/100; //converts centivolts into volts
```

```
void setup() {           // This function gets called when the Arduino starts
  Serial.begin(115200);  // This code sets up the Serial port at 115200 baud
  rate (bps)
  pinMode(ledPin, OUTPUT); // Set the digital pin as output
}
//Function called Thermister to perform the fancy math of the Steinhart-Hart
equation:
double Thermister(int RawADC) {
  double Temp;           // Create the internal variable Temp (only inside the
function)
  Temp = log(((10240000/RawADC) - 10000));
  Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp
);
  Temp = Temp - 273.15;           // Convert Kelvin to Celsius
  //Temp = (Temp * 9.0)/ 5.0 + 32.0; // uncomment for Celsius to Fahrenheit
  return Temp;
}

void loop() {           //This function loops while the arduino is powered
//Read the temperature from the sensor and store it in temp:
  int val;              //Create an integer variable
  double temp;          //Variable to hold a temperature value
  val=analogRead(0);    //Read the analog port 0 and store the value in val
  temp=Thermister(val); //Runs the fancy math on the raw analog value

//Compare temperature with target and decide:
  //if temperature is too high, turn the led off and increase the power supplied
to the Peltiers
  if(temp >= Tmax) {
    ledState = LOW;
    power += 0.5;
    //if temperature is too low, turn the led off and decrease the power supplied
to the Peltiers
  }else if(temp < Tmin){
    ledState = LOW;
    power -= 0.5;
  }
}
```

```
//if temperature is within the limits, turn on the LED and mantain the power
supply
  }else{
    ledState = HIGH;
  }
if(power > 99) power = 99; //Sets power to 99 if it surpasses the upper limit
if(power < 0) power = 0; //Sets power to 0 if it surpasses the lower limit
peltier_level = map(power, 0, 99, 0, 255);
centivolt = map(power, 0, 99, 0, 500);
voltage = centivolt/100.00;

// Monitor Serie printing:
Serial.print ("Target:");
Serial.print (TargetTemp);
Serial.print ("  Temp:");
Serial.print (temp);
Serial.print ("  LED:");
Serial.println (ledState);
Serial.print("Power=");
Serial.print(power);
Serial.print("%");
Serial.print("  Voltage=");
Serial.print(voltage);
Serial.println("V");
Serial.println (".");

//Send information to the Peltier port and the LED
digitalWrite(ledPin, ledState); //Set the LED with the ledState of the
variable
analogWrite(peltier, peltier_level); //Write this new value out to the port
delay(2000); //Wait 2 seconds before we do it again
}
```