import RPi.GPIO as GPIO

#Import SPI Library (for hardware SPI) and MCP3008 Library

import Adafruit\_GPIO.SPI as SPI

import Adafruit\_MCP3008

import time

import math

#Hardware SPI configuration

SPI\_PORT = 0

SPI\_DEVICE = 1

mcp = Adafruit\_MCP3008.MCP3008(spi = SPI.SpiDev(SPI\_PORT,SPI\_DEVICE))

#Blink 10 times the LED with on/off intervals of 500ms

#Pin numbers equal to board numbers

GPIO.setmode(GPIO.BOARD)

#Make pin 11 an output

GPIO.setup(11, GPIO.OUT)

while True:

for j in range(10):

GPIO.output(11, True)

time.sleep(0.5)

GPIO.output(11, False)

time.sleep(0.5)

#Read the photoresistor and the thermistor

#Read the photoresistor first

rawPhotoresistor = mcp.read\_adc(0)

#Read the termistor second

rawThermistor = mcp.read\_adc(1)

#Display the photoresistor value (raw)

print “The photoresistor’s raw value is” + str(rawPhotoresistor) + “\n”

#Display the temperature value in Kelvin

#Value in Kelvin

referenceTemp = 298

#Setting a random value for the resistance at the reference temperature

referenceResistance = 1000

#Given value in the handout

B = 3435

#Values used to calculate the termistor’s temperature

voltage = (rawThermistor / 1023)\*(3.3)

#Using the value of a 1 kOhm pull down resistor, according to the lab handout

current = voltage / 1000

resistance = (3.3 – voltage) / current

#Temperature of thermistor in Kelvin

temperature = (1.0 / referenceTemp) + ((1.0 / B)\*math.log(resistance/ referenceResistance))

#Using formula provided in the handout

thermistorTemp = 1/temperature

#Concatenating the message into a string

print “Thermistor (Kelvin) ” + “ = ” + str(thermistorTemp) + “\n”

#Display the temperature value in Celsius so the user can interpret it better

celsiusTemp = thermistorTemp – 273.15

print “The thermistor’s temperature in Celisus is ” + “ = ” + str(celsiusTemp) + “\n”

#Pause for 15 seconds so the loop doesn’t run that fast

time.sleep(15)