

## Senzori analogici

Perifericele dintr-un sistem cu procesor pot comunica

- Analogic
- Digital

cu partea centrala a sistemului (microcontroller, microprocesor).

In aceasta lucrare vom studia cateva exemple de interactiune cu perifericele tip senzor, intr-un sistem compatibil Arduino UNO.

Se poate folosi ca referinta suplimentara cursul Robofun, <https://www.robofun.ro/docs/curs/8764634/CursGratuitArduino-Lectia2-SenzoriAnalogici.pdf> cu observatia ca in aceasta lucrare vom folosi alti senzori decat cei existenti la vanzare de robofun.

**Observatie: Toate modificarile de conexiune pe breadboard din acest laborator se efectueaza fara placa Arduino alimentata.**

### Senzor de temperatura NTC – 3 pini

Se foloseste un senzor NTC cu 3 pini (VCC, GND, DO).

- Linia VCC este pentru alimentarea la 5V Arduino
- GND, pentru legarea la masa Arduino



Un astfel de senzor detecteaza temperaturi in gama 20°C - 80°C, si le traduce in voltaj aplicat intre bornele – si +.

Pinul DO returneaza rezultatul comparatiei temperaturii masurate, convertita in volti, cu un prag (reglat prin potentiometru). Pentru comparatie se foloseste un comparator tip LM393.

### Temperatura (digital)

Pentru aceasta lucrare vom avea nevoie de

- Arduino uno
- Fire tata-tata
- Breadboard

1.1. Se insereaza senzorul de temperatura in breadboard. Se realizeaza matricea de conexiune senzor – Arduino de mai jos.

Pin modul senzor	Pin Arduino
VCC	5V
GND	GND
DO	D2

1.2. Se programeaza Arduino.

Programul urmatore foloseste pinul D2 al Arduino pentru a citi valoarea DO de la senzor, si aprinde LED-ul in functie de pragul de temperatura. Aceasta este o aplicatie tipica de detector de caldura.

```
const int sensorDOPin = 2;    // connect DO pin to pin 2 of Arduino
const int ledPin = 13;       // the number of the LED pin

// variables will change:
int SensorState = 0;         // variable for the state of the sensor

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the sensor pin DO as input
  pinMode(sensorDOPin, INPUT);
}

void loop() {
  // read the state of the sensor value:
  SensorState = digitalRead(sensorDOPin);
  // by default, the sensor sends a HIGH all the time
  //if the sensor sends a LOW
  if (SensorState == LOW) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

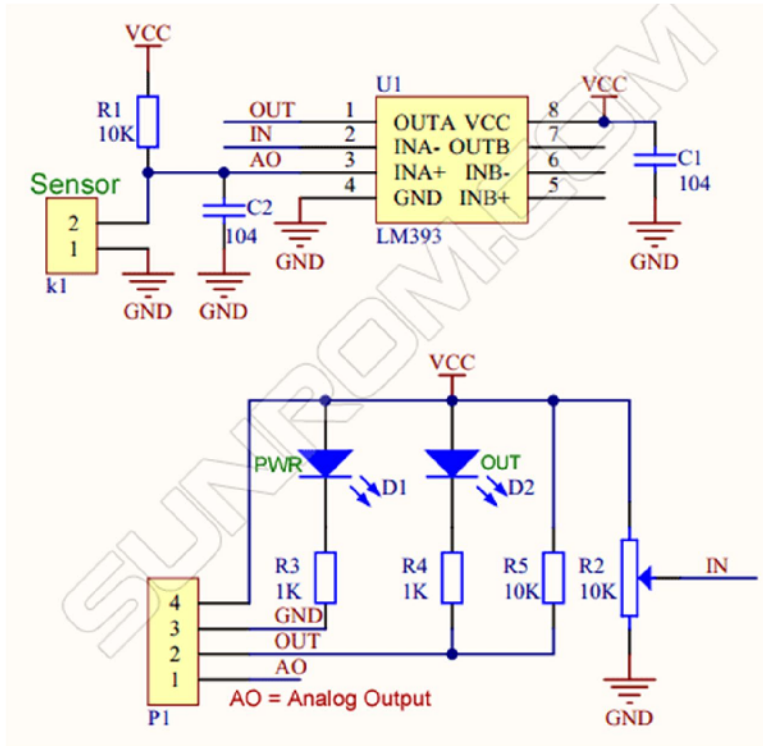
1.3. Cum ar trebui modificat programul pentru a trimite la portul serial, pe 9600 baud, informatia de DO?

### Temperatura (analogic)

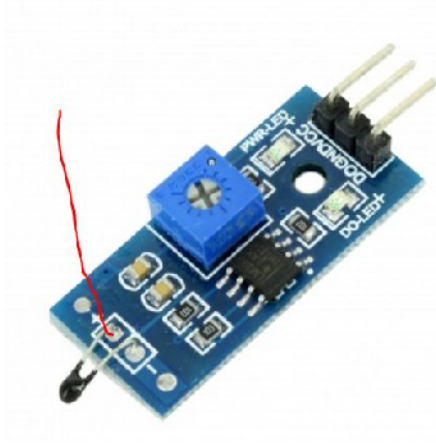
In mod normal ar trebui sa folosim un senzor de temperatura cu 4 pini (VCC, GND, DO, AO), si sa accesam linia AO- analog out, a modului.

2.1. Cum, din nefericire, avem la dispozitie doar un senzor cu 3 pini, va trebui sa identificam punctul de achizitie pentru marimea analogica (semnalul AO).

Schema electrica tipica a acestui senzor este cea de mai jos.



Se observa ca marimea AO se achizitioneaza de la terminalul pozitiv al termistorului.



De data aceasta avem nevoie de a conecta terminalul + al termistorului la Arduino, pe un pin capabil de ADC. Pinul D2, de la punctul precedent, este capabil de samplare ADC.

2.2. Se realizeaza matricea de conexiune.

Pin modul senzor	Pin Arduino
VCC	5V
GND	GND
DO	---
AO	D2

2.3. Se ruleaza programul.

```

#define LED_PIN 13           // define on-board LED
#define DIGITAL_IN_PIN 3    // define the digital input pin
#define ANALOG_IN_PIN A0   // define the analog input pin

int    digitalValue ; // read digital value
int    analogValue;   // read analog value
int    correctedAnalogValue;// Value corrected for "Upside down" Thermistor
float  celsiusTemperature;

void setup()  /***** SETUP: RUNS ONCE *****/
{
  pinMode (LED_PIN, OUTPUT) ;      // Onboard LED
  pinMode (DIGITAL_IN_PIN, INPUT) ;// digital input signal (Not actually required; INPUT is
  default)
  pinMode (ANALOG_IN_PIN, INPUT) ;// analog input signal (Not actually required; INPUT is
  default)
  Serial.begin(9600);              // Start the Serial Monitor connection
  Serial.println("Arduino UNO: Thermistor Analog-Digital Brick Data:");
} //--(end setup )---

void loop()   /***** LOOP: RUNS CONSTANTLY *****/
{
  /*-----( Read the Analog data, do calculations )-----*/
  analogValue = analogRead(ANALOG_IN_PIN);
  Serial.print("Analog value = ");
  Serial.print(analogValue), DEC; // display analog value

  // The Brick has the resistor-thermistor positions reversed from what the
  // "Thermistor" algorithm expects. "Fix It In Software" :-)
  correctedAnalogValue = map(analogValue, 0, 1023, 1023, 0);

  celsiusTemperature = Thermistor(correctedAnalogValue); // Calculate temperature

  /*-----( Read the Digital data, do calculations )-----*/
  digitalValue = digitalRead (DIGITAL_IN_PIN) ;
  Serial.print(" Digital value = ");
  Serial.println(digitalValue), DEC; // display digital value
  Serial.print("High Temperature Switch = ");
  if (digitalValue == HIGH) // When the sensor value exceeds the set point, LED is turned on
  {
    digitalWrite (LED_PIN, HIGH);
    Serial.print("ON ");
  }
  else
  {
    digitalWrite (LED_PIN, LOW);
    Serial.print("OFF ");
  }
  /*-----( Print out the calculated Temperatures )-----*/
  Serial.print("Measured Temperature = ");
  Serial.print(celsiusTemperature, 1); // display Celsius
  Serial.println(" C");

  Serial.println(); // A blank line separator
  delay(1000);      // Wait a little
} //--(end main loop )---

/*-----( Declare User-written Functions )-----*/
// "Thermistor" function does a 3rd order formula to find the temperature from the voltage
// See http://en.wikipedia.org/wiki/Thermistor
double Thermistor(int RawADC)
{
  double Temp; // Working variable
  Temp = log(((10240000 / RawADC) - 10000));
  Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
  //Cube!
  Temp = Temp - 273.15;          // Convert Kelvin to Celsius
  return Temp;
} // END Thermistor Function

```