MERENJE TEMPERATURE U VIŠE TAČAKA POMOĆU ARDUINO MODULA

MEASURING TEMPERATURE IN SEVERAL POINTS USING ARDUINO

Nada RATKOVIĆ KOVAČEVIĆ[1], Sanja JEVTIĆ[2], Đorđe DIHOVIČNI[3], Dragan KRECULJ[4], Marija Zajeganović[5], Luka KOCIĆ[6]

[1] The Academy of Applied Technical Studies Belgrade, Department of Computer-Mechanical Engineering,
[2] Academy of Technical and Art Applied Studies Belgrade, Department School of Railway transport,
[3] The Academy of Applied Technical Studies Belgrade, Department of Computer-Mechanical Engineering,
[4] The Academy of Applied Technical Studies Belgrade, Department of Computer-Mechanical Engineering,
[5] Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies,
[6] The Academy of Applied Technical Studies Belgrade, Department of Computer-Mechanical Engineering

Merenje temperature ili termometrija opisuje proces merenja trenutne lokalne temperature za istovremenu ili kasniju procenu. Temperatura je veoma važna i često merena promenljiva, od značaja za mnoge inženjerske primene, bilo u mašinstvu ili u elektrotehnici. Različite su primene, koje zahtevaju da se u nekom periodu odredi trenutna temperatura lokalno, to jest, meri se: telesna temperatura (za otkrivanje bolesti), temperatura vazduha (za podešavanje rada HVAC sistema – sistema za grejanje, ventilaciju i klimatizaciju), zagrevanje električnih kontakata ili elektronskih komponenti, temperatura zavarivanja, prati se i reguliše temperatura u procesnoj industriji, prati se temperatura na mrežnim uređajima i serverima, ...

Električni gubici (ili gubici druge vrste energije) često su u obliku toplote.

Ovde napravljen prototip, koristi Arduino Uno za prikupljanje i obradu podataka u okviru merenja temperature. Temperatura se meri u tri tačke, pomoću digitalnih senzora DS18B20. Digitalni senzori ovog tipa imaju individualne identifikatore, što omogućava umrežavanje uređaja uz mogućnost zasebnog očitavanja svake od temperatura. Protokol, koji se koristi za komunikaciju između Arduino Uno modula i senzora temperature DS18B20 je 1-wire protokol, deo vlasničke Maxim Integrated 1-Vire® tehnologije. Ovaj protokol obezbeđuje serijsku komunikaciju i nekoliko senzora može da bude i jeste, povezano na isti pin Arduino Uno.

Nakon što je prototip uređaja sastavljen, programira se pomoću Arduino IDE softverskog sistema i modifikovanih rešenja - skica (eng. sketches), dostupnih u okviru Arduino zajednice. Izveden je pokušaj je da se kalibriše ovaj merni uređaj, merenjem temperature kocke leda i ključale vode. Ovde prikazan prototip mernog uređaja je izrađen u okviru pristupa obrazovanju, zasnovanom na praksi, koji se koristi na akademijama strukovnih studija.

Ključne reči: *mikrokontroleri; obrazovanje zasnovano na praksi i projektovanju; senzori; merenje temperature; umrežavanje uređaja*

Temperature Measurement or Thermometry describes a process of measuring current local temperature for immediate or future evaluation. Temperature is a very critical and widely measured variable for many of the engineering application, either mechanical or electrical ones. Variety of applications exhibit necessity to determine local temperature in time, i. e. measurement of: the body temperature (to detect illness), air temperature (to tune-up the HVAC – Heating, Ventilation and Air Conditioning system operation), electrical contacts or electronic components heating, welding *temperature, temperature monitoring and control in process industry, the temperature on network devices and servers is monitored, etc.*

Electrical (or other type of) energy losses are often in the form of heat.

Here a prototype is built using Arduino Uno for data acquisition and data processing within temperature measurement. The temperature is measured in three points using digital sensors DS18B20. Digital sensors of this type have individual identifiers, which enables networking of devices while being able to read each of the temperatures separately. The protocol used for communication between Arduino Uno module and DS18B20 temperature sensors is 1-wire protocol, part of proprietary Maxim Integrated 1-Wire® technology. This protocol provides serial communication and several sensors can be and are connected to the same pin of Arduino Uno.

After a prototype of the device is built it is programmed using Arduino IDE software system and modification of sketches available in Arduino community. An attempt is made to calibrate this measuring device, by measuring the temperature of the ice cube and boiling water. The prototype of the measuring device is being built within the project based education performed in academies of vocational studies.

Key words: *microcontrollers; networking devices; project based education; sensors; temperature measurements*

1 Introduction

Temperature is believed to be the most widely measured variable [1]. Temperature is a measure of molecular energy, or heat energy, and the potential to transfer heat energy [1].

Many processes in everyday life or in industry must have either a monitored or controlled temperature [2]. This can range from the checking out the body temperature, the monitoring water temperature of an engine or load device, or being as complex as the temperature of a weld in a laser welding application. Measuring and controlling the air temperature is ubiquitous in tune-up of the HVAC – Heating, Ventilation and Air Conditioning systems [2].

In the process control of chemical reactions, temperature control is of major importance, since chemical reactions are temperature-dependent [1]. All physical parameters are temperature-dependent, making it necessary in most cases to measure temperature along with the physical parameter, so that temperature corrections can be made to achieve accurate parameter measurements [1].

More difficult measurements may need to be monitored such as the temperature of smoke stack gas from a power generating station or blast furnace or the exhaust gas of a rocket [2]. Much more common are the temperatures of fluids in processes or process support applications, or the temperature of solid objects such as metal plates, bearings and shafts in a piece of machinery. [2]

Instrumentation also can be temperature-dependent, requiring careful design or temperature correction, which can determine the choice of measurement device [1].

The importance of accurate temperature measurement cannot be overemphasized [1].

2 Hardware

2.1 Arduino Uno

Arduino is a name of a project, a range of open-source hardware platforms having microcontrollers for physical programming and a community of people being experts or novices interested in making and programming electronic devices on their own [3], [4]. Arduino [3] is an open-source electronics development platform based on easy-to-use hardware and software. Arduino boards are able to read digital or analog inputs and to produce digital or PWM (Pulse Width Modulated) outputs. Arduino boards can 'listen' to a sensor, read a finger pressed on a button, or read either a SMS (Short Message Service) or Twitter message - and turn it into an output - activating a motor, turning on an LED (Light Emitting Diode), publish something online [3].

Arduino Uno is selected to be used here, since it is easy to program and enables developing of electronic projects like fun and play. Arduino Uno is a very popular hardware platform and is often in use in practical experimenting on various levels of education [3], [4]. Arduino Uno has a microcontroller ATmega328P with 6 analog inputs and 14 digital inputs/outputs, which can be software defined. Some of these 14 pins can be defined as PWM (Pulse Width Modulated) digital outputs providing the control of devices requiring analog inputs. The prototype of electronic circuit and corresponding sketch – program presented here is adapted from existing examples, built for measuring temperature using either analog [5], [6] or digital temperature sensors [7], or both [8], [9], [10].

2.2 Digital temperature sensor DS18B20

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points [11]. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. The DS18B20 can be powered directly from the data line ("parasite power"), eliminating the need for an external power supply [11].

One microprocessor can control many DS18B20 sensors distributed over a large area [11]. Each sensor of this type has a unique 64-bit serial code, which allows multiple DS18B20 chips to communicate with processor over the same 1-Wire bus. Applications that can benefit from this feature include [11]: HVAC environmental controls; temperature monitoring systems inside buildings, equipment, or machinery; process monitoring; and control systems [11].

Another feature of the DS18B20 is the ability to operate without an external power supply [11]. Power is instead supplied through the 1-Wire pullup resistor through the DQ pin when the bus is high. The high bus signal also charges an internal capacitor, which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." [11] As an alternative, the DS18B20 may also be powered by an external supply on VDD and this is recommended if temperatures above 100°C are to be measured. [11]

Prototype of electronic circuit having one, two or three DS18B20 sensors is taken after and adapted from [7]. In [7] a device with one DS18B20 sensor is derived. Similar to [7], various examples of circuits for temperature measurement using single DS18B20 sensors are given in [8] - [10].

A completed prototype of electronic circuit is shown in Figure 1. with three DS18B20 temperature sensors (having 3-Pin TO-92 package types) connected to the Arduino Uno module.

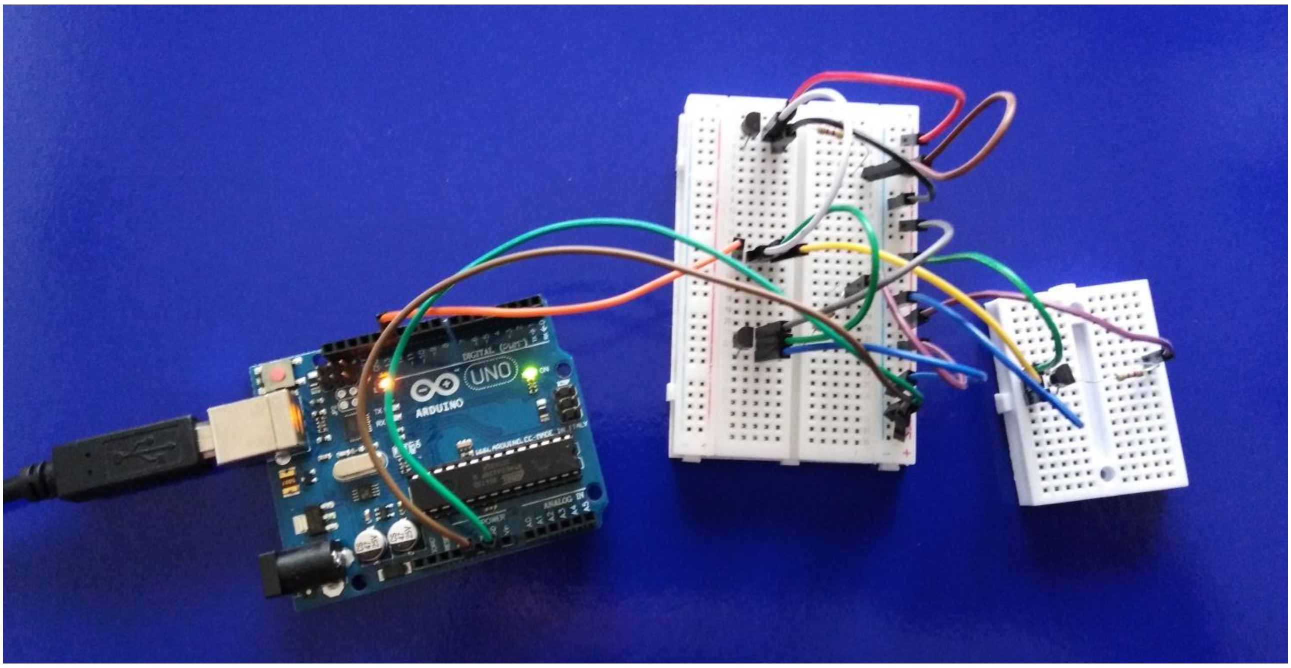

Figure 1. The prototype of the electronic circuit with Arduino Uno communicating with the three DS18B20 temperature sensors

3 Software

3.1 Arduino IDE – Integrated Development Environment

One can tell to the Arduino unit what to do by sending a set of instructions to the microcontroller on the board. To do so one can use the Arduino programming language (based on Wiring), and the Arduino Software - Arduino Integrated Development Environment (Arduino IDE), based on Processing [3], [4]. Programs written for Arduino modules are called sketches.

In order to make Arduino Uno communicate using 1-wire protocol and read temperature measurements from a DS18B20 sensors, it was necessary to expand Arduino IDE with two libraries, as suggested in e. g. [7]: DallasTemperature.h (a Miles Burton's library, [12] - [14]) and OneWire.h (a Paul Stoffregen's library, [15], [16]). Adding libraries to Arduino IDE is made user-friendly and the process is explained in [7], [12] - [16]. Arduino enthusiasts more competent in coding can write libraries and import these to the Arduino IDE. A list of currently available community contributed libraries for Arduino can be found on [17].

The sketch for this Arduino Uno example is taken from [18] (e.g. [7]) and slightly adapted to suit calling/ polling the three sensors instead of one (serial communication speed had to be adapted as well, as suggested in [7]):

```
// DSThreeTempSensor_sketch_jul20c

// Adapted from sketch published on the site
// https://github.com/akarsh98/DS18B20-ESP8266-Arduino/blob/master/Basic%20Code.ino

// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into port 12 on the Arduino
#define ONE_WIRE_BUS 12

// libraries configuration
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

/*
 * The setup function. We only start the sensors here
 */
void setup(void)
{
  // start serial port
  //Serial communication speed had to be adapted  //Serial.begin(115200);
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
}

/*
 * Main function, get and show the temperature
```

```
 */
void loop(void)
{
  // Setup a oneWire instance to communicate with any OneWire devices
  // (not just Maxim/Dallas temperature ICs)
  OneWire oneWire(ONE_WIRE_BUS);

  // Pass our oneWire reference to Dallas Temperature.
  DallasTemperature sensors(&oneWire);

  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  // After we got the temperatures, we can print them here. We use the function ByIndex,
  // and as an example get the temperature from the first sensor only.
  Serial.print("Temperature for the device 1 (index 0) is: ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.println("°C ");
  Serial.print("Temperature for the device 2 (index 1) is: ");
  Serial.print(sensors.getTempCByIndex(1));
  Serial.println("°C ");
  Serial.print("Temperature for the device 3 (index 2) is: ");
  Serial.print(sensors.getTempCByIndex(2));
  Serial.println("°C ");

  delay(1000);

}
```

Figure 2. shows a print screen of the sketch in operation (on the right) with temperature measurements displayed in Serial Monitor application (on the left). The sketch is opened and translated in Arduino IDE, and the loop() function of the sketch is visible on the right side of the Figure 2.

4 1-wire bus system and protocol

The 1-Wire bus system uses a single bus master to control one or more slave devices [11]. The DS18B20 sensor is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system. If there are multiple slaves on the bus the system is "multidrop". All data and commands are transmitted least significant bit first over the 1-Wire bus. [11]

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal [11]. The control line requires a weak pullup resistor (approximately of 5 kΩ) since all devices are linked to the bus via a 3-state or open-drain port. In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code.

The transaction sequence for accessing the DS18B20 has three steps. Their order is as follows [11]: Step 1 - Initialization; Step 2 - ROM Command (followed by any required data exchange); and Step 3 - DS18B20 Function Command (followed by any required data exchange). It is very important to follow this sequence every time the DS18B20 is accessed, as the sensor will not respond if any steps in the sequence are missing or out of order [11]. Exceptions to this rule are - after issuing either of these ROM commands, Search ROM or Alarm Search, the master must return to Step 1 in the sequence. [11]

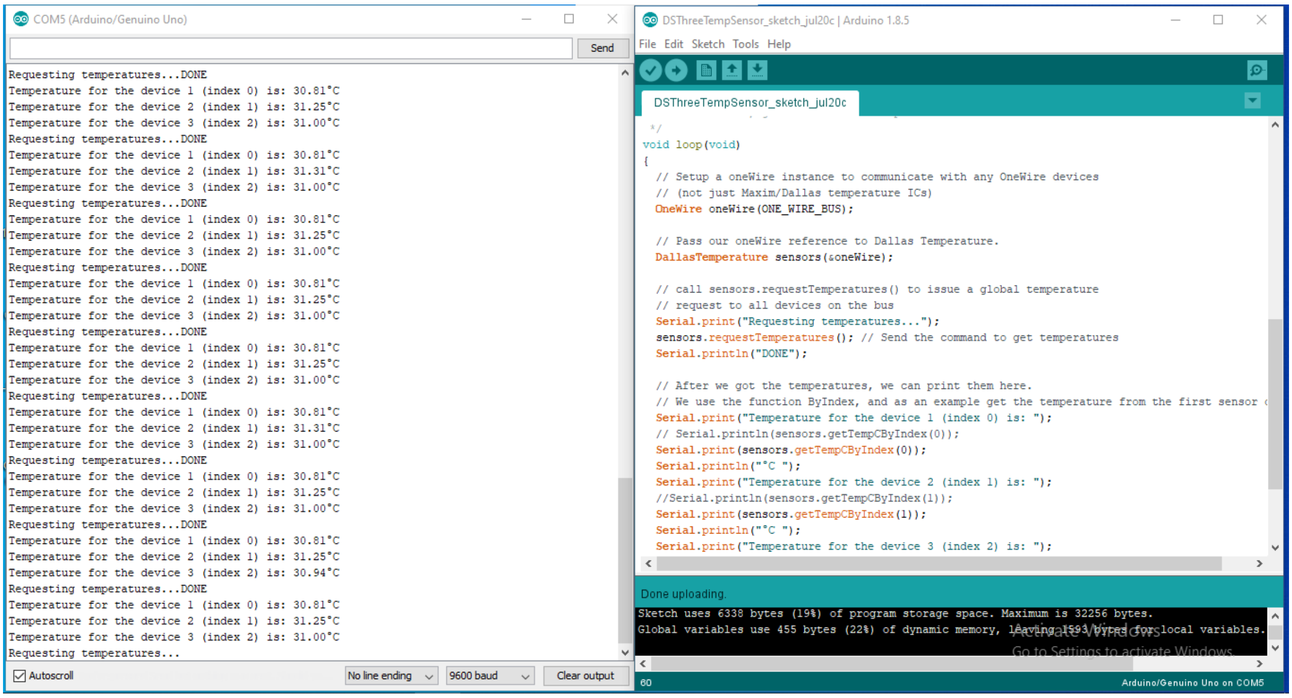

Figure 2. The print screen showing the sketch (opened in Arduino IDE on the right) in operation with temperature measurements displayed in Serial Monitor application (on the left)

5 Calibration

In order to make use of the measuring device it has to be calibrated. Most temperature-sensing devices are rugged and reliable, but they can go out of calibration, due to leakage during use or contamination during manufacture, and therefore should be checked on a regular basis [1].

Temperature calibration can be performed on most temperature sensing devices by immersing them in known temperature standards, which are the equilibrium points of solid/liquid or liquid/gas mixtures [1]. Some of these standards or temperature scale calibration points include [1]:

- water: solid-liquid mixture, providing reference for 0°C, and
- water: liquid-gas mixture, providing reference for 100°C.

For the calibration a waterproof type of DS18B20 is used. Calibration is performed or rather attempted on one sensor. Since temperature sensor used here has “parasite power” it may be risky to try to measure 100°C. This point for calibration would require external power supply for the DS18B20. Calibration was attempted for 0°C, and lowest measured temperature was 2.19°C, as shown in Fig. 3. Due to insufficient amount of ice, calibration was not achieved.

Contact measurement of the temperature can be erroneous due to a heat transfer between the sensor and the environment which could cause changes in temperature. Nevertheless contact measurement of the temperature is still in use nowadays.

6 Conclusion

Temperature measurement or monitoring is very important. Contact measurement of the temperature can be made using DS18B20 digital sensor. Using 1-wire bus and protocol many DS18B20 sensors can be connected to a single line to a master processor device. Here, a master device used is Arduino Uno board.

Digital sensor DS18B20 has software programmable resolution. Alarms can be defined in software too. This sensor is available in several packaging, one of them being waterproof which is useful if sensor is supposed to be submerged in liquid. DS18B20 can be used to measure and/or control temperature in various real life applications (e.g. HVAC systems, process control, server equipment monitoring and protection from overheating, …).

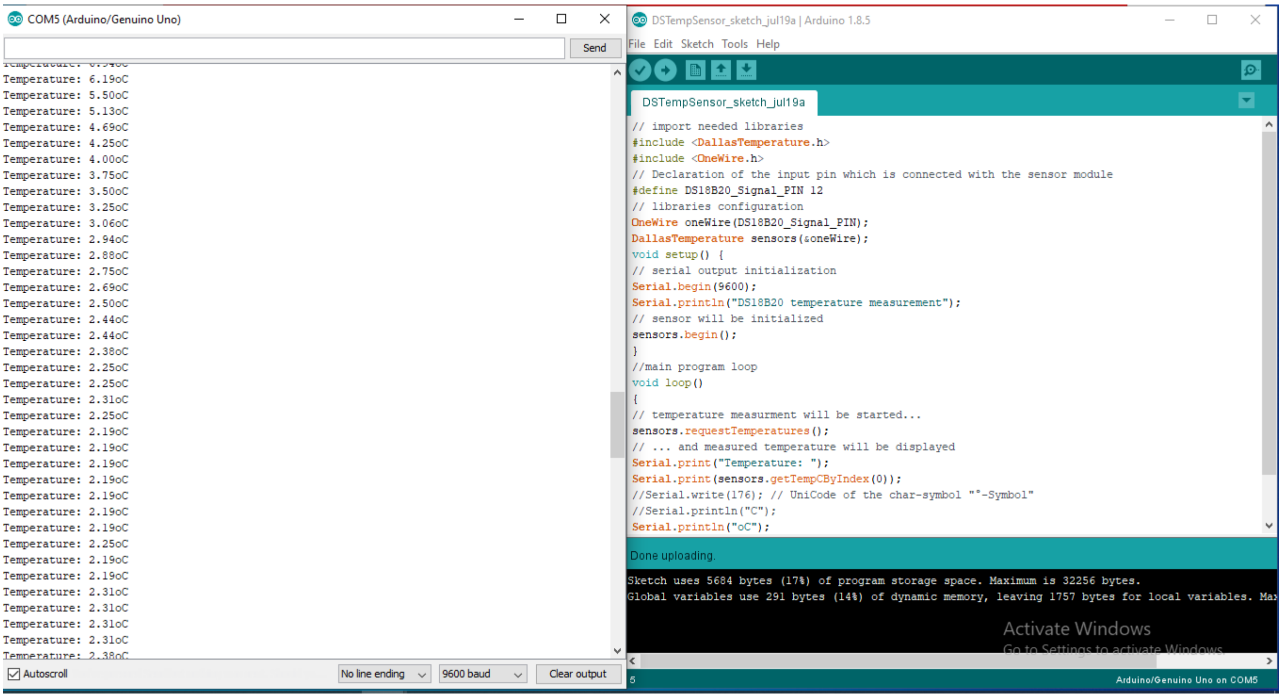

Figure 3. The print screen showing the sketch (opened in Arduino IDE on the right) in operation with temperature measurements displayed in Serial Monitor application (on the left)

Here a prototype device was built having three DS18B20 sensors which communicate over 1-wire bus and protocol with an Arduino Uno board. One DS18B20 of waterproof type was used in an attempt to perform calibration of sensor. The endeavor displayed here was performed within a project based education in vocational studies. There is room for further improvement of the device.

7 References

[1] **Dunn, W. C.,** *Introduction to Instrumentation, Sensors, and Process Control*, Artech House Sensors library, Boston & London, USA, 2006.

[2] **Cottrell, M. S.,** Temperature Measurement, Mechanical Instrumentation (On-line Resources for the ME240 course), University of Missouri – Rolla, 2006. [Published on-line on 17. 08. 2006.] (accessed 29. 07. 2022.) https://web.mst.edu/~cottrell/ME240/Resources/Temperature/Temperature.pdf

[3] ***, *Introduction*, *Arduino foundation web site*, Arduino Foundation, [Published on-line Last revision February 05, 2018.], 2018. https://www.arduino.cc/en/Guide/Introduction (accessed on 25. 07. 2022.)

[4] **Ratković Kovačević, N., Đ. Dihovični, D. Kreculj, I. Krulj, R. Stefanović, D. Milovanović**, *Examples of Internet of Things - IoT Implementations in Education*, Proceedings *27th Conference and Exhibition YuINFO 2021*, Društvo za informacione sisteme i računarske mreže, Zlatibor, Serbia, 2021. pp. 140-145. (accessed 25. 07. 2022.) http://www.yuinfo.org/ZBORNIK_YU_INFO_2021.pdf

[5] a) ***, *SparkFun Inventor's Kit Guidebook - V3.3*, version 3.3, Contributors HelloTechie and Toni_K, SparkFun, (accessed 29. 07. 2022.) https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33
b) ***, *SIK Guide - Your guide to the SparkFun Inventor's Kit for the SparkFun RedBoard*, version 3.3, SparkFun, (accessed 29. 07. 2022.) https://cdn.sparkfun.com/datasheets/Kits/SIK/V33/SIK%203.3%20Manual.pdf

[6] a) ***, SparkFun Inventor's Kit Experiment Guide - v4.1, Contributors Joel_E_B and BBoyHo e.g. Ho Yun Bobby Chan, SparkFun, (accessed 29. 07. 2022.)

https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v41
b) ***, *SparkFun Inventor's Kit version 4.0a - Your Guide to the SIK for the SparkFun RedBoard*, SparkFun, (accessed 29. 07. 2022.)
https://github.com/sparkfun/SIK_Guide/raw/master/English/SIK%20v4.1%20Book%202019%20WEB.pdf

[7] **Agarwal, A.,** Temperature Sensors for Arduino ESP8266 & ESP31 - DS18B20 Temperature Sensor Tutorial with Arduino and ESP8266, Arduino Foundation, [Published on-line on June 7 2020.] (accessed 30. 07. 2022.)
https://create.arduino.cc/projecthub/akarsh98/ds18b20-temperature-sensor-tutorial-with-arduino-and-esp8266-db31aa

[8] **Dogan, I.,** The Ultimate Compendium of Sensor Projects: 40+ Projects using Arduino, Raspberry Pi and ESP32, Elektor International Media BV / for translation to Serbian Agencija Eho, Niš, Serbia, 2019.

[9] ***, *SENSORKIT X40*, JOY-IT, (accessed 29. 07. 2022.)
a) https://www.joy-it.net/en/products/SEN-Kit02 b) https://sensorkit.joy-it.net/en/
c) https://www.joy-it.net/files/files/Produkte/SEN-Kit02/SEN-KitX40-Datasheet.pdf

[10] **Santos, R.,** 9 Arduino Compatible Temperature Sensors for Your Electronics Projects, Random Nerd Tutorials, (accessed 30. 07. 2022.)
https://randomnerdtutorials.com/9-arduino-compatible-temperature-sensors-for-your-electronics-projects/

[11] ***, *DS18B20 Programmable Resolution 1-Wire Digital Thermometer, Data Sheets,* Maxim Integrated Products Inc., 19-7487; Rev 6; 2019. [Published on-line on July 2019]
https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf (accessed 30. 07. 2022.)

[12] **Burton, M.,** *Arduino Library for Maxim Temperature Integrated Circuits*, GitHub, [Published on-line], (accessed 22. 08. 2022.)
https://github.com/milesburton/Arduino-Temperature-Control-Library

[13] **Burton, M.,** *Dallas Temperature Control Library*, [Published on-line and last modified on 15. 01. 2016.], (accessed 22. 08. 2022.)
https://www.milesburton.com/Dallas_Temperature_Control_Library

[14] **Burton, M.,** *Arduino Library for Dallas Temperature ICs Supports DS18B20, DS18S20, DS1822, DS1820*, Arduino Foundation, [Published on-line], (accessed 22. 08. 2022.)
https://www.arduino.cc/reference/en/libraries/dallastemperature/

[15] *****,** *Dallas Semiconductor's 1-Wire Protocol*, Arduino Playground web page, Arduino Foundation web site, [Published on-line], (accessed 30. 08. 2022.)
https://playground.arduino.cc/Learning/OneWire/

[16] *****,** *OneWire Library*, PJRC Electronic Projects Components Available Worldwide, [Published on-line], (accessed 30. 08. 2022.)
https://www.pjrc.com/teensy/td_libs_OneWire.html

[17] ***, *Libraries for Arduino*, Arduino Playground web page, Arduino Foundation web site, [Published on-line], (accessed 30. 08. 2022.)
https://playground.arduino.cc/Main/LibraryList/

[18] **GitHub User akarsh98,** *DS18B20-ESP8266-Arduino*, GitHub, [Published on-line on 30 May 2020], (accessed 22. 08. 2022.)
https://github.com/akarsh98/DS18B20-ESP8266-Arduino/blob/master/Basic%20Code.ino