



# INTERNET PAMETINIH UREĐAJA

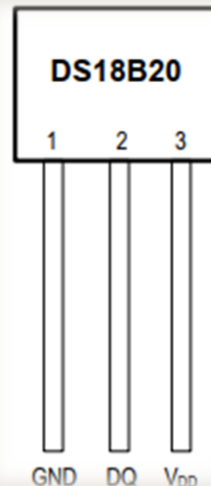
prof. dr Dejan S. Aleksić

Prirodno-matematički fakultet, Niš

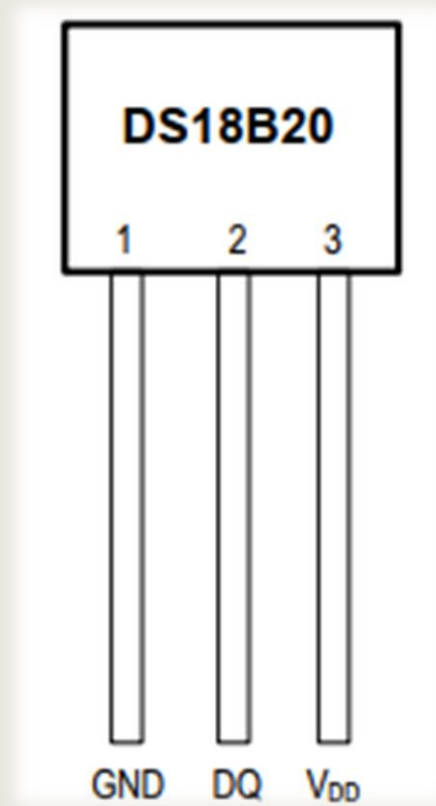
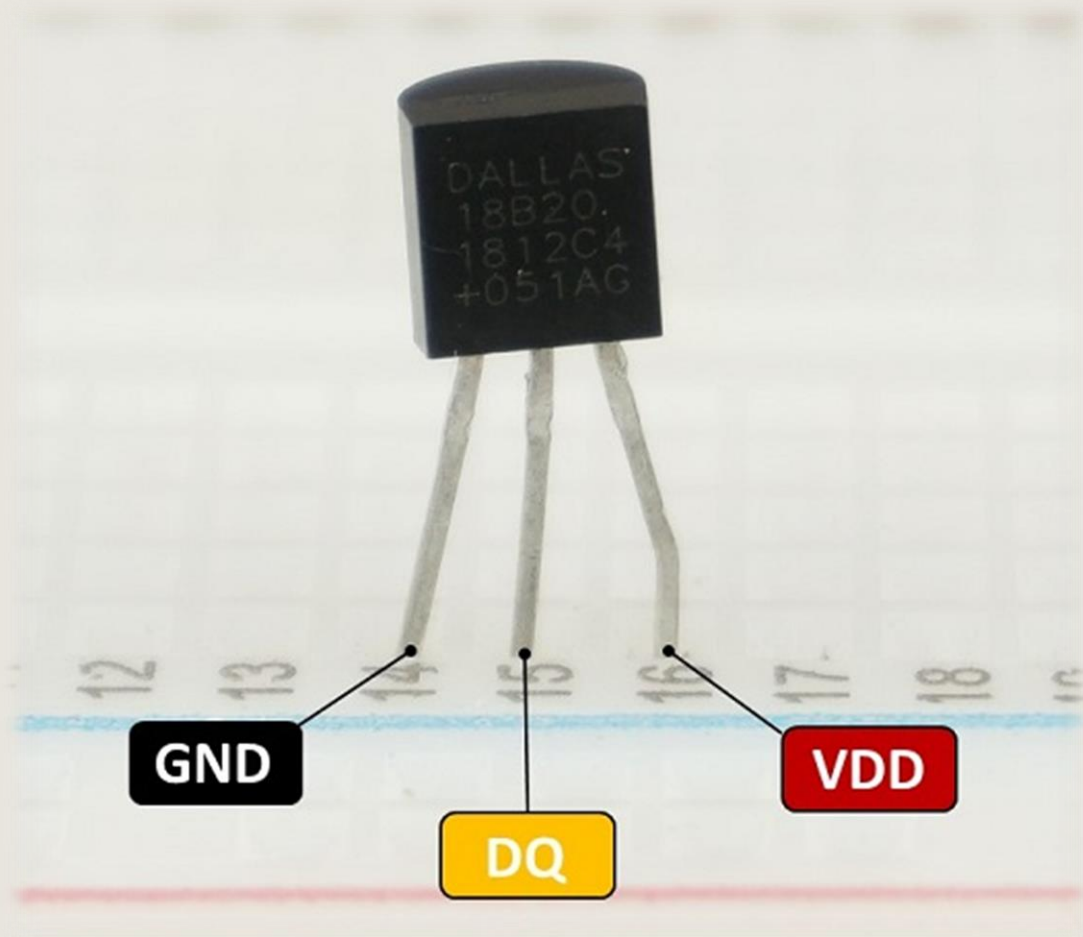
## 02. TEMPERATURNI SENZOR DS18B20

# TEMPERATURNI SENSOR DS18B20

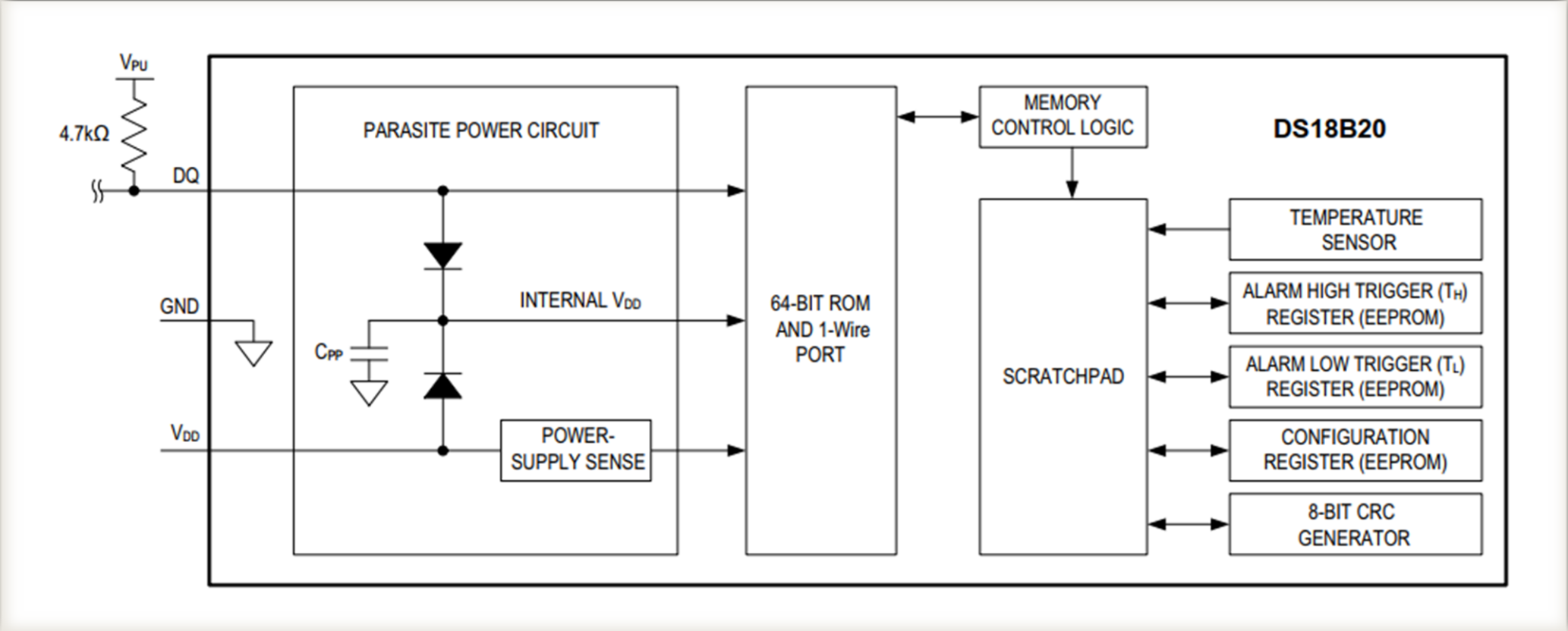
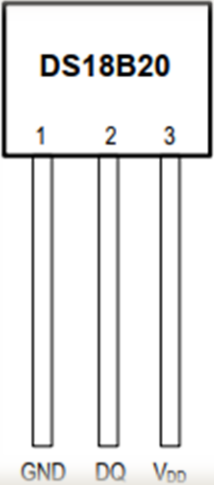
- Digitalni termometar DS18B20 obezbeđuje 9-bitna do 12-bitna merenja temperature
- Merni opseg od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$
- Preciznost  $\pm 0.5^{\circ}\text{C}$  za opseg temperature od  $-10^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$
- Postoji mogućnost programiranja alarma
- Komunikacija sa DS18B20 se obavlja preko 1-Wire magistrale koja po definiciji zahteva samo jednu liniju podataka (i masu) za komunikaciju sa  $\mu$ kontrolerom
- Moguće je napajanje direktno sa linije podataka („parasite power“), eliminišući potrebu za spoljnim napajanjem.
- Svaki DS18B20 ima jedinstveni 64-bitni serijski kod – adresu čime se omogućava da više DS18B20 funkcionišu na istom 1-Wire magistrali
- Primanjuje se u HVAC, praćenje temperature unutar zgrada, opreme ili mašina, u sistemima za praćenje i kontrolu procesa, ...



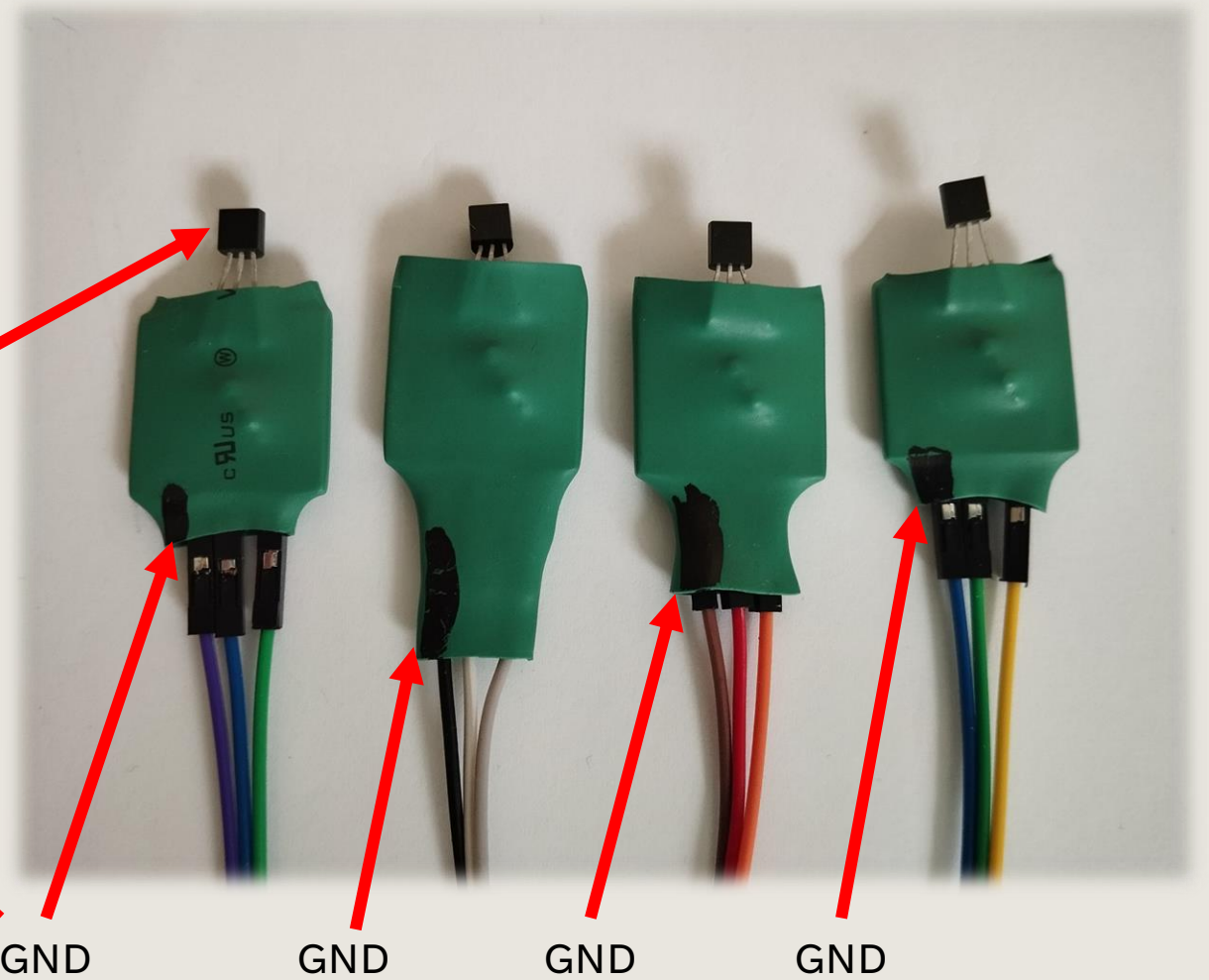
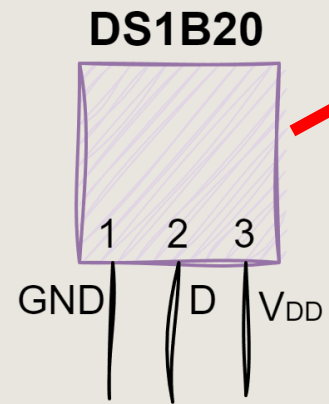
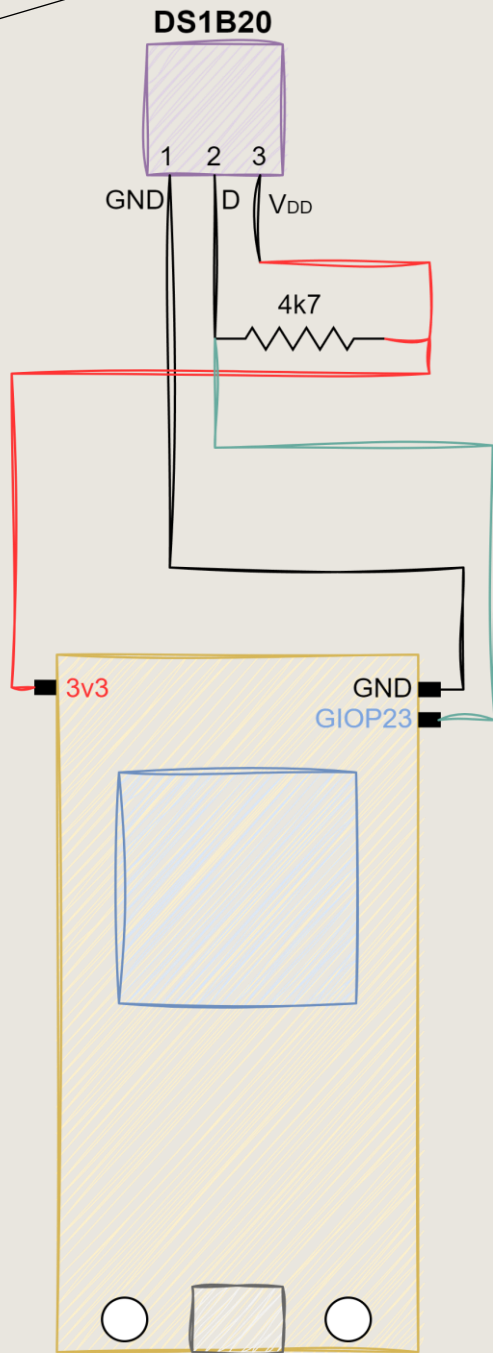
# TEMPERATURNI SENSOR DS18B20



# TEMPERATURE SENSOR DS18B20



# ŠEMA POVEZIVANJA



# TRAZENJE ADRESE SENZORA

```
#include <Arduino.h>

#include <OneWire.h>

OneWire ds(23);

void setup(void) {

    Serial.begin(115200);

    Serial.printf ("Setup finised.\n");
}
```

# TRAZENJE ADRESE SENZORA

```
void loop(void) {
  byte i;
  byte present = 0;
  byte data[12];
  byte addr[8];

  ds.reset_search();

  if (!ds.search(addr)) {
    Serial.print("No more addresses.\n");
    ds.reset_search();
    return;
  }

  Serial.print("R=");
  for( i = 0; i < 8; i++) {
    Serial.print(addr[i], HEX);
    Serial.print(" ");
  }
}
```

# TRAZENJE ADRESE SENZORA

```
if ( OneWire::crc8(addr, 7) != addr[7]) {
    Serial.print("CRC is not valid!\n");
    return;
}

if (addr[0] == 0x10) {
    Serial.print("Device is a DS18S20 family device.\n");
}
else {
    if (addr[0] == 0x28) {
        Serial.print("Device is a DS18B20 family device.\n");
    }
    else {
        Serial.print("Device family is not recognized: 0x");
        Serial.println(addr[0],HEX);
        return;
    }
}
```



```
ds.reset();
ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at the end

delay(1000);      // maybe 750ms is enough, maybe not

present = ds.reset();
ds.select(addr);
ds.write(0xBE);   // Read Scratchpad

Serial.print("P=");
Serial.print(present,HEX);
Serial.print(" ");

for ( i = 0; i < 9; i++) {    // we need 9 bytes
  data[i] = ds.read();
  Serial.print(data[i], HEX);
  Serial.print(" ");
}

Serial.print(" CRC=");
Serial.print (OneWire::crc8(data, 8), HEX);
Serial.println();
}
```

# TRAZENJE ADRESE SENZORA

```
Serial.print(" CRC=");  
  
Serial.print (OneWire::crc8( data, 8), HEX);  
  
Serial.println();  
}
```

# ČITANJE TEMPERATURE 2

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int SENSOR_PIN = 23; // pin connected to DS18B20 sensor's DQ pin

OneWire oneWire(SENSOR_PIN);          // setup a oneWire instance
DallasTemperature tempSensor(&oneWire); // pass oneWire to DallasTemperature library

float tempCelsius;    // temperature in Celsius
float tempFahrenheit; // temperature in Fahrenheit
```

# TRAZENJE ADRESE SENZORA

```
void setup()
{
  Serial.begin(9600); // initialize serial
  tempSensor.begin(); // initialize the sensor
}

void loop()
{
  tempSensor.requestTemperatures(); // send the command to get temperatures
  tempCelsius = tempSensor.getTempCByIndex(0); // read temperature in Celsius
  tempFahrenheit = tempCelsius * 9 / 5 + 32; // convert Celsius to Fahrenheit

  Serial.print("Temperature: ");
  Serial.print(tempCelsius); // print the temperature in Celsius
  Serial.print("°C");
  Serial.print(" ~ "); // separator between Celsius and Fahrenheit
  Serial.print(tempFahrenheit); // print the temperature in Fahrenheit
  Serial.println("°F");

  delay(500);
}
```