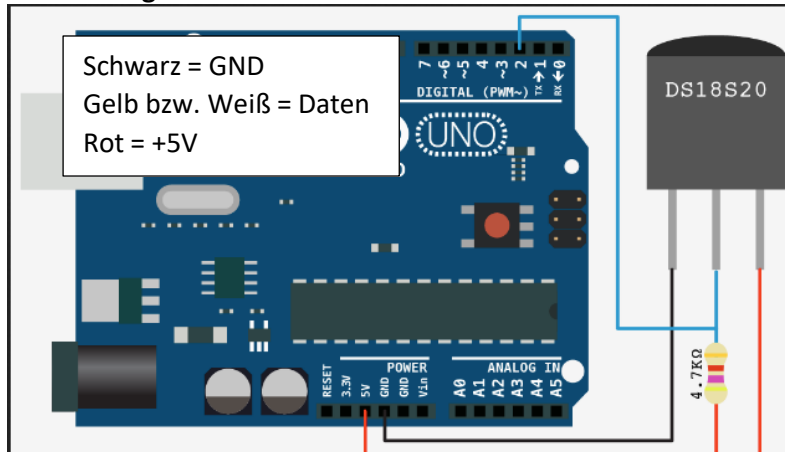


**Beschreibung:**

Der große Vorteil ist, dass man den DS1820 problemlos in Reihe schalten kann – somit werden keine zusätzlichen Pins belegt.

**Beschaltung:****Bild:****Sketch:**

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2 // Datenpin 2
OneWire oneWire(ONE_WIRE_BUS); // Richtet eine oneWire-Instanz ein
DallasTemperature sensors(&oneWire); // Gibt die oneWire-Referenz an Dallas Temperature weiter.
void setup(void)
{
  Serial.begin(9600); //Begin serial communication
  Serial.println("Arduino Digital Temperature // Serial Monitor Version"); //Print a message
  sensors.begin();
}
void loop(void)
{
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.print("Temperature is: ");
  Serial.println(sensors.getTempCByIndex(0)); //Ausgabe der Temperatur für den ersten Sensor
  //Serial.print(sensors.getTempCByIndex(1)); //Ausgabe der Temperatur für den zweiten Sensor usw.
  //alle Sensoren werden parallel geschaltet
  delay(1000);
}
```

**Bibliothek: OneWire-master.zip und DallasTemperature-3.7.6.zip**

Jeder 1-Wire Chip enthält eine 64-bit Seriennummer im 'rom', bestehend aus einer 8-bit Kennung für den Schaltkreistyp, einer 48-bit Seriennummer, und eine 8-bit CRC. Diese CRC wird verwendet um die Integrität der Daten sicherzustellen.

Um einen HEX Wert in einen Analogwert zu konvertieren muss als erstes geklärt werden ob es sich um einen DS18S20, oder DS18B20 handelt. Der Code um die Temperatur zu errechnen, muss geringfügig für den DS18B20 (und DS1822) angepasst werden, weil dieser einen 12-bit Temperaturwert (0.0625 °C Auflösung) liefert, während der DS18S20 und der DS1820 einen 9-bit Wert liefern (0.5 °C Auflösung).