ID S3

Function and characterization of a temperature and humidity sensor with 1-wire interface and a gas sensor with I2C interface

Required Knowledge:

- 1-wire Interface
- Environmental Measurements: Temperature, Humidity, VOC's
- Arduino Programming
- Python Programming (Anaconda/Spyder)
- Datasheet DHT-11 (https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf or https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf)
- Datasheet CCS811 (e.g. https://cdn.sparkfun.com/assets/learn_tutorials/1/4/3/CCS811_Datasheet-DS000459.pdf)

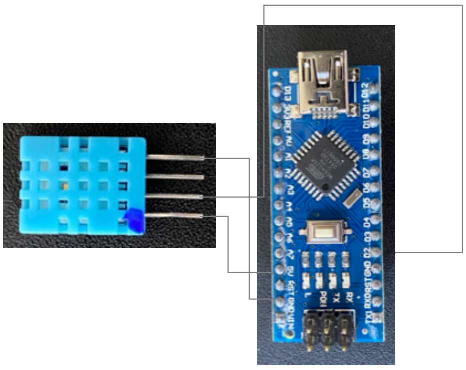
Material:

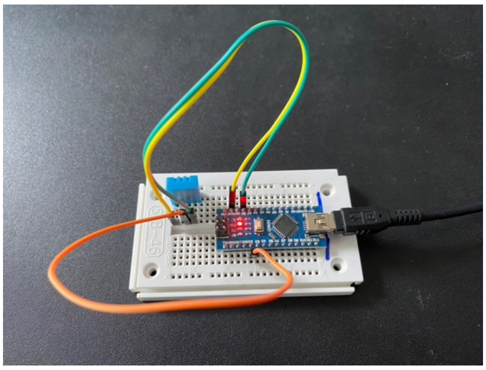
- Arduino Nano
- Temperature and Humidity Sensor DHT11
- Breakout Board CCS811
- Breadboard
- USB-Mini Cabel
- 3 Male-Male Connectore (1-wire)
- 5 Male-Female Connectors (CCS811 / I2C)
- Laptop, Arduino-IDE, Python-IDE (z.B. Anaconda), simple terminal software (z.B. HTerm)

Duration: approx. 4h

Setup:

Hardware:





Please note: According to the specification, a 10kOhm resistor should also be used as a pull-up between signal line and 5V. The sensor then pulls the line to ground with low resistance for communication.

WARNING: Reverse polarity (5V/GND) will destroy the sensor (a hole will appear on the back). Thus, make sure, the sensor is connected correctly BEFORE turning the power on!

Software:

- Arduino IDE with the libraries „DHT sensor library" and „Adafruit Unified Sensor"
- Anaconda/Spyder
- Prepared programmes from the Teams site (Arduino, Python)

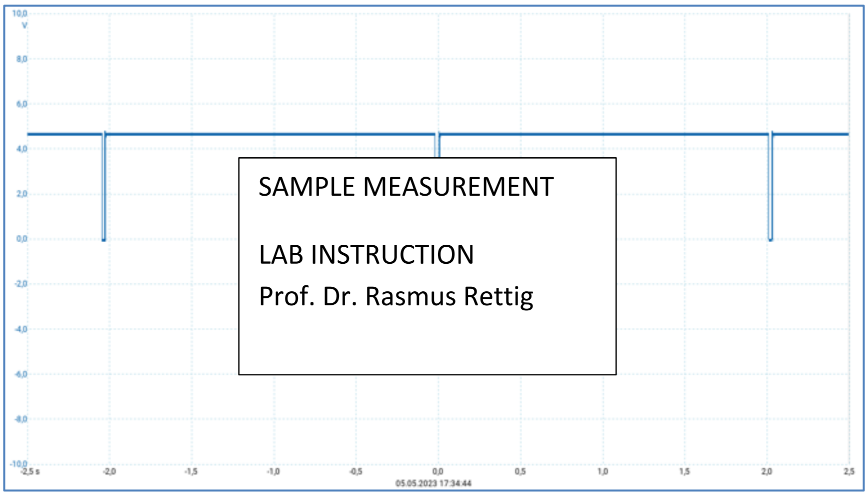
Part 1 - Setup

- Setup the Arduino and the DHT11 on the breadboard
- Three connections are required: 5V, GND, 1-wire pin (in code: Arduino / D2); **MAKE SURE, THE DHT11 IS CONNECTED CORRECTLY BEFORE POWERING ON (NO REVERSE POLARITY)**

- Download the program "Basisprogramm_DHT11_Arduino" from the teams site, compile it and transfer it to the Arduino
- Check the transmitted times with "Tools/Serial Monitor" or "Tools/Serial Plotter"; make sure that the transmission parameters (COM port and baud rate) are set correctly
- Download the "Basisprogramm_ReadUSB_DHT11" program and load it into your editor under Anaconda/Spyder
- Start the program there and check the output values. You may have to install the "pyserial", "numpy" and "matplotlib" libraries in the Anaconda Navigator beforehand
- Make sure that the COM port and baud rate are also set correctly in the program
- Analyze both programs: Which measurement data is recorded and how does the measurement data get onto the laptop, include a brief explanation and flow chart for the data flow

Part 2 – Analysis of the physical layer of the 1-wire interface with an oscilloscope

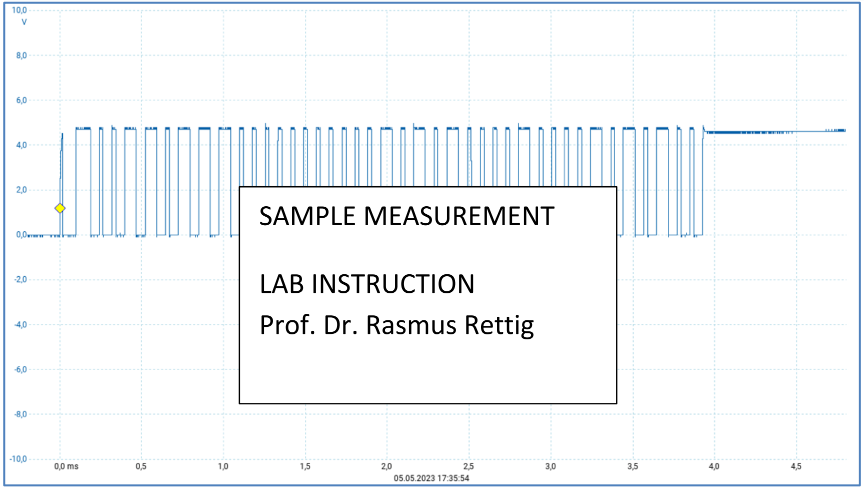
- Connect the oscilloscope to your measurement setup. Measure the voltage between the 1-wire line and GND with a probe (please adjust the square-wave signal beforehand!).
- Examine the 1-wire protocol: How does a telegram look like?
  How often are messages sent per second?
- Decode your device ID based on your measurement. Document the result in your lab report!



SAMPLE MEASUREMENT

LAB INSTRUCTION

Prof. Dr. Rasmus Rettig



SAMPLE MEASUREMENT

LAB INSTRUCTION

Prof. Dr. Rasmus Rettig

Part 3 – Sensor time constants

- Set up the sensor and let it run until the values for temperature and humidity no longer change (typically a few minutes).
- Start the measurement data acquisition in Python

- Breathe briefly on the sensor, watch the temperature and humidity rise and then fall again. Save and plot your measurement result RH(t) and T(t). Plot with the y-axis both linearly and logarithmically scaled.
- Fit an exponential function to the decreasing curves of both quantities. To do this, select a suitable area of the measurement curves. Which time constants τ do you observe?
- Confirm your results by plotting both the measured data and the exponential function into the same graph.

Part 4 – Automatic fan control

- Modify the Arduino program so that the built-in LED (pin 13) turns on when the temperature exceeds a threshold. If the threshold value is not reached again, the LED should be switched off again. This function could also be used to switch on a fan via a relay or power transistor.

Part 5 – Integration of CCS811

- Figure out how to integrate the CCS811 into the system (hardware, software), document your approach and implement it
- In each run of the program, in addition to the measurement data from the DFT11, also transfer the data from the CCS811 and store and save them together
- Carry out a measurement over about 5 minutes and record the following five measured values in parallel: DHT11: temperature, relative humidity; CCS811: Temperature (please measure and correct the offset beforehand), TVOC, CO2
- Compare all measurement data graphically with Python and describe/discuss the relationships (5 coordinate systems with a common time axis)