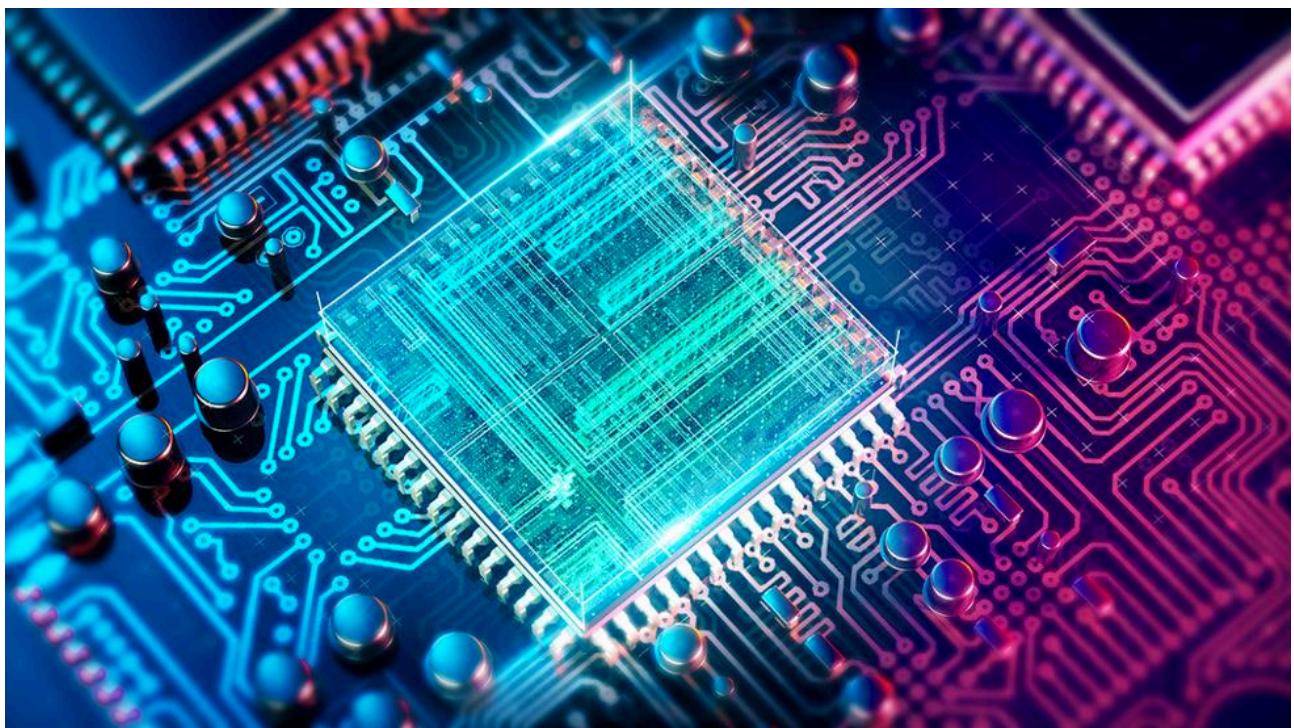


Dokumentation

Abschlussprojekt

Embedded Systems

Prof. Dr.-Ing. Juergen Schuele



von Christian Kolb und Patric Richardson
Hochschule für Gestaltung Schwäbisch Gmünd
Wintersemester 2019/2020
Internet der Dinge

Aufgabenstellung:

Logger für Umweltdaten:

Speichert Zeitstempel, Druck-, Temperatur- und Umgebungslichtdaten in EEPROM
EEPROM-Auslesung.

Auf Knopfdruck: csv-Daten an die serielle Schnittstelle senden
Auswertung der Daten.

Erzeugen Sie ein Diagramm aus csv-Daten mit einem Werkzeug Ihrer Wahl, z. B. Excel, Python.

Die Daten müssen mindestens 24 Stunden umfassen.

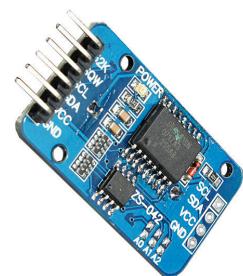
Komponenten:



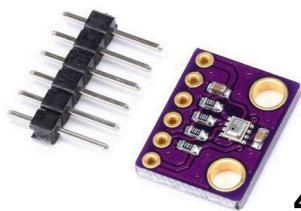
1



2



3



4



5



6

1. EEPROM
2. NUCLEO-F091RC
3. DS3231 Real Time Clock
4. BMP-280 Temperatur und Luftdrucksensor
5. Phototransistor
6. Widerstände

Schaltung:

Beschaltung der Pins am Nucleo-F091RC:

- 3V3
- GND
- Analog Pin 0
- PWM/D9
- PWM/MOSI/D11
- MISO/D12
- SDA/D14
- SCL/D15

Beschaltung der Pins am BMP280:

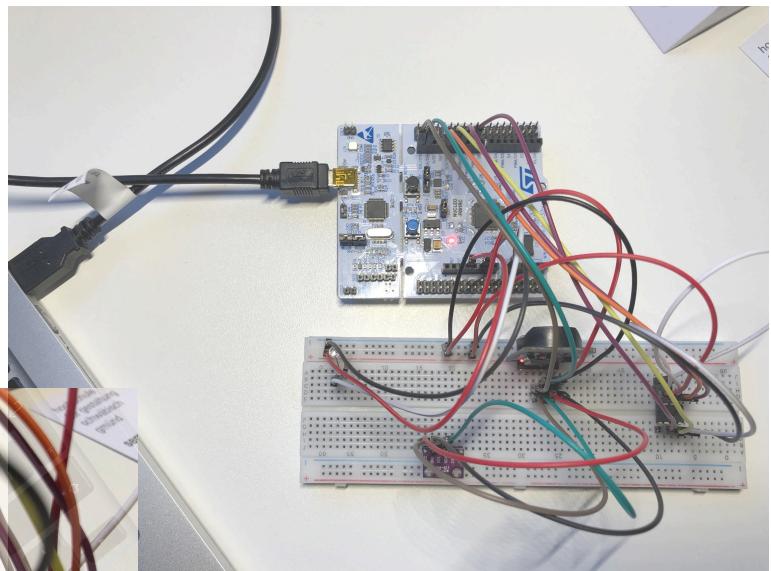
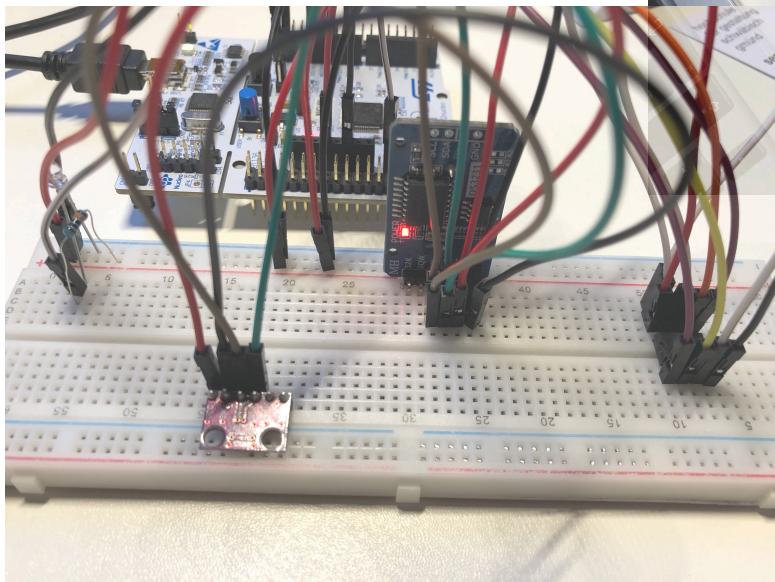
- GND
- VCC
- SDA
- SCL

Beschaltung der Pins am DS3231:

- VCC
- GND
- SCL
- SDA

Beschaltung der Pins an DS3231:

- VCC
- GND
- SCL
- SDA



Terminal-Ausgabe:

```
chriss-MacBook-Pro-2 ~ % screen /dev/tty.usbmodem14203 115200 -L > SCREEN — 109x50
chris — screen /dev/tty.usbmodem14203 115200 -L
Embedded Systems Projekt Christian Kolb und Patric Richardson
Beginn der Messung
Anzahl Auswertung: 1
```

screen /dev/tty.usbmodem14203 115200 -L, öffnet Programm um Messungen zu beginnen.

```
chriss-MacBook-Pro-2 ~ % screen /dev/tty.usbmodem14203 115200 -L > SCREEN — 109x50
10:40:0.8.2.20.0.0,45533
10:41:0.8.2.20.0.0,45403
10:42:0.8.2.20.0.0,45307
10:43:0.8.2.20.0.0,45083
10:44:0.8.2.20.0.0,45771
10:45:0.8.2.20.0.0,45211
10:46:0.8.2.20.0.0,45243
10:47:0.8.2.20.0.0,46011
10:48:0.8.2.20.0.0,45851
10:49:0.8.2.20.0.0,45050
10:50:0.8.2.20.0.0,45739
10:51:0.8.2.20.0.0,45595
10:52:0.8.2.20.0.0,45563
10:53:0.8.2.20.0.0,45851
10:54:0.8.2.20.0.0,45659
10:55:0.8.2.20.0.0,45435
10:56:0.8.2.20.0.0,45563
10:57:0.8.2.20.0.0,45483
10:58:0.8.2.20.0.0,45355
10:59:0.8.2.20.0.0,42154
11:0:0.8.2.20.0.0,45195
11:1:0.8.2.20.0.0,44986
11:2:0.8.2.20.0.0,45307
11:3:0.8.2.20.0.0,45419
11:4:0.8.2.20.0.0,45355
11:5:0.8.2.20.0.0,45355
11:6:0.8.2.20.0.0,45403
11:7:0.8.2.20.0.0,45307
11:8:0.8.2.20.0.0,45083
11:9:0.8.2.20.0.0,45771
11:10:0.8.2.20.0.0,45211
11:11:0.8.2.20.0.0,45243
11:12:0.8.2.20.0.0,46011
11:13:0.8.2.20.0.0,45851
11:14:0.8.2.20.0.0,45050
11:15:0.8.2.20.0.0,45739
11:16:0.8.2.20.0.0,45595
11:17:0.8.2.20.0.0,45563
11:18:0.8.2.20.0.0,45851
11:19:0.8.2.20.0.0,45659
11:20:0.8.2.20.0.0,45435
11:21:0.8.2.20.0.0,45563
Anzahl Auswertung: 1342
```

Ausschnitt der Messungen am 8.2.20 mit 1341 Auswertungen, Gesamt Anzahl der Auswertungen/Messungen: 1441

Problem BMP-280:

Leider konnten die Werte von dem BMP280 Sensor für Luftdruck und Temperatur nicht ausgelesen werden, es wurde mit Codes nur für den BMP280 getestet und es wurde versucht den BMP280 mit einem Arduino und passenden Skript (im Anhang) auszulesen dies führte leider nicht zu dem gewünschten Erfolg. Um trotzdem ein Diagramm für Temperatur und Luftdruck zu erstellen wurden plausible Werte verwendet. Diese Werte sind nicht von dem BMP280 Sensor gekommen.
Die Programmierung wurde in Zusammenarbeit mit der HS-Aalen gemacht.

Ausschnitte Messwerte als CSV:

Erfasste Daten im Bereich von 7.2.20, 11:21:0 bis 8.2.20, 11:21:0

Anzahl der Auswertungen: 1441

| | | |
|--------------------------|--------------------------|--------------------------|
| 11:21:0,7.2.20,0,0,45403 | 12:2:0,7.2.20,0,0,45483 | 10:39:0,8.2.20,0,0,45355 |
| 11:22:0,7.2.20,0,0,45307 | 12:3:0,7.2.20,0,0,45355 | 10:40:0,8.2.20,0,0,45355 |
| 11:23:0,7.2.20,0,0,45083 | 12:4:0,7.2.20,0,0,42154 | 10:41:0,8.2.20,0,0,45403 |
| 11:24:0,7.2.20,0,0,45771 | 12:5:0,7.2.20,0,0,45195 | 10:42:0,8.2.20,0,0,45307 |
| 11:25:0,7.2.20,0,0,45211 | 12:6:0,7.2.20,0,0,44986 | 10:43:0,8.2.20,0,0,45083 |
| 11:26:0,7.2.20,0,0,45243 | 12:7:0,7.2.20,0,0,45307 | 10:44:0,8.2.20,0,0,45771 |
| 11:27:0,7.2.20,0,0,46011 | 12:8:0,7.2.20,0,0,45419 | 10:45:0,8.2.20,0,0,45211 |
| 11:28:0,7.2.20,0,0,45851 | 12:9:0,7.2.20,0,0,45355 | 10:46:0,8.2.20,0,0,45243 |
| 11:29:0,7.2.20,0,0,45050 | 12:10:0,7.2.20,0,0,45355 | 10:47:0,8.2.20,0,0,46011 |
| 11:30:0,7.2.20,0,0,45739 | 12:11:0,7.2.20,0,0,45403 | 10:48:0,8.2.20,0,0,45851 |
| 11:31:0,7.2.20,0,0,45595 | 12:12:0,7.2.20,0,0,45307 | 10:49:0,8.2.20,0,0,45050 |
| 11:32:0,7.2.20,0,0,45563 | 12:13:0,7.2.20,0,0,45083 | 10:50:0,8.2.20,0,0,45739 |
| 11:33:0,7.2.20,0,0,45851 | 12:14:0,7.2.20,0,0,45771 | 10:51:0,8.2.20,0,0,45595 |
| 11:34:0,7.2.20,0,0,45659 | 12:15:0,7.2.20,0,0,45211 | 10:52:0,8.2.20,0,0,45563 |
| 11:35:0,7.2.20,0,0,45435 | 12:16:0,7.2.20,0,0,45243 | 10:53:0,8.2.20,0,0,45851 |
| 11:36:0,7.2.20,0,0,45563 | 12:17:0,7.2.20,0,0,46011 | 10:54:0,8.2.20,0,0,45659 |
| 11:37:0,7.2.20,0,0,45483 | 12:18:0,7.2.20,0,0,45851 | 10:55:0,8.2.20,0,0,45435 |
| 11:38:0,7.2.20,0,0,45355 | 12:19:0,7.2.20,0,0,45050 | 10:56:0,8.2.20,0,0,45563 |
| 11:39:0,7.2.20,0,0,42154 | 12:20:0,7.2.20,0,0,45739 | 10:57:0,8.2.20,0,0,45483 |
| 11:40:0,7.2.20,0,0,45195 | 12:21:0,7.2.20,0,0,45595 | 10:58:0,8.2.20,0,0,45355 |
| 11:41:0,7.2.20,0,0,44986 | 12:22:0,7.2.20,0,0,45563 | 10:59:0,8.2.20,0,0,42154 |
| 11:42:0,7.2.20,0,0,45307 | 12:23:0,7.2.20,0,0,45851 | 11:0:0,8.2.20,0,0,45195 |
| 11:43:0,7.2.20,0,0,45419 | 12:24:0,7.2.20,0,0,45659 | 11:1:0,8.2.20,0,0,44986 |
| 11:44:0,7.2.20,0,0,45355 | 12:25:0,7.2.20,0,0,45435 | 11:2:0,8.2.20,0,0,45307 |
| 11:45:0,7.2.20,0,0,45355 | 12:26:0,7.2.20,0,0,45563 | 11:3:0,8.2.20,0,0,45419 |
| 11:46:0,7.2.20,0,0,45403 | 12:27:0,7.2.20,0,0,45483 | 11:4:0,8.2.20,0,0,45355 |
| 11:47:0,7.2.20,0,0,45307 | 12:28:0,7.2.20,0,0,45355 | 11:5:0,8.2.20,0,0,45355 |
| 11:48:0,7.2.20,0,0,45083 | 12:29:0,7.2.20,0,0,42154 | 11:6:0,8.2.20,0,0,45403 |
| 11:49:0,7.2.20,0,0,45771 | 12:30:0,7.2.20,0,0,45195 | 11:7:0,8.2.20,0,0,45307 |
| 11:50:0,7.2.20,0,0,45211 | 12:31:0,7.2.20,0,0,44986 | 11:8:0,8.2.20,0,0,45083 |
| 11:51:0,7.2.20,0,0,45243 | 12:32:0,7.2.20,0,0,45307 | 11:9:0,8.2.20,0,0,45771 |
| 11:52:0,7.2.20,0,0,46011 | 12:33:0,7.2.20,0,0,45419 | 11:10:0,8.2.20,0,0,45211 |
| 11:53:0,7.2.20,0,0,45851 | 12:34:0,7.2.20,0,0,45355 | 11:11:0,8.2.20,0,0,45243 |
| 11:54:0,7.2.20,0,0,45050 | 12:35:0,7.2.20,0,0,45355 | 11:12:0,8.2.20,0,0,46011 |
| 11:55:0,7.2.20,0,0,45739 | 12:36:0,7.2.20,0,0,45403 | 11:13:0,8.2.20,0,0,45851 |
| 11:56:0,7.2.20,0,0,45595 | 12:37:0,7.2.20,0,0,45307 | 11:14:0,8.2.20,0,0,45050 |
| 11:57:0,7.2.20,0,0,45563 | 12:38:0,7.2.20,0,0,45083 | 11:15:0,8.2.20,0,0,45739 |
| 11:58:0,7.2.20,0,0,45851 | 12:39:0,7.2.20,0,0,45771 | 11:16:0,8.2.20,0,0,45595 |
| 11:59:0,7.2.20,0,0,45659 | 12:40:0,7.2.20,0,0,45211 | 11:17:0,8.2.20,0,0,45563 |
| 12:0:0,7.2.20,0,0,45435 | 12:41:0,7.2.20,0,0,45243 | 11:18:0,8.2.20,0,0,45851 |
| 12:1:0,7.2.20,0,0,45563 | 12:42:0,7.2.20,0,0,46011 | 11:19:0,8.2.20,0,0,45659 |
| | 12:43:0,7.2.20,0,0,45851 | 11:20:0,8.2.20,0,0,45435 |
| | | 11:21:0,8.2.20,0,0,45563 |

Diagramm Vergleich Temperatur und Luftdruck:

(plausible Werte)

Daten von 7.02.20, 11:21:0 bis 7.02.20, 12:21:0

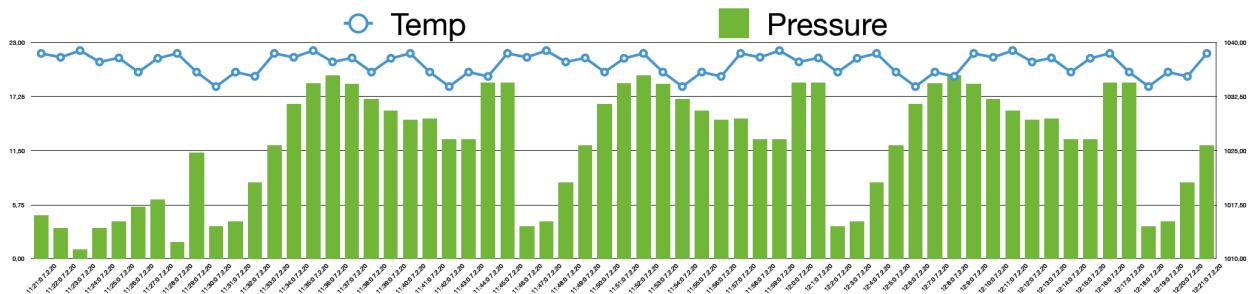


Diagramm Lichtverhältnisse:

Daten von 7.2.20, 11:21:0 bis 7.2.20, 12:21:0

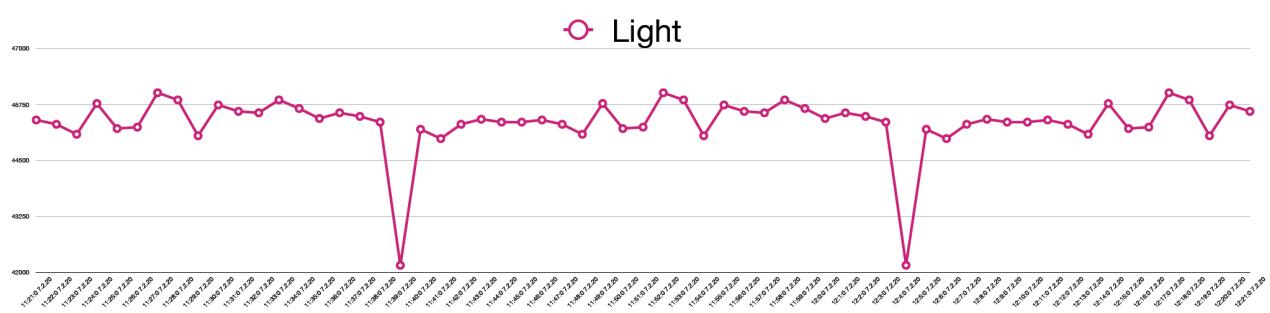
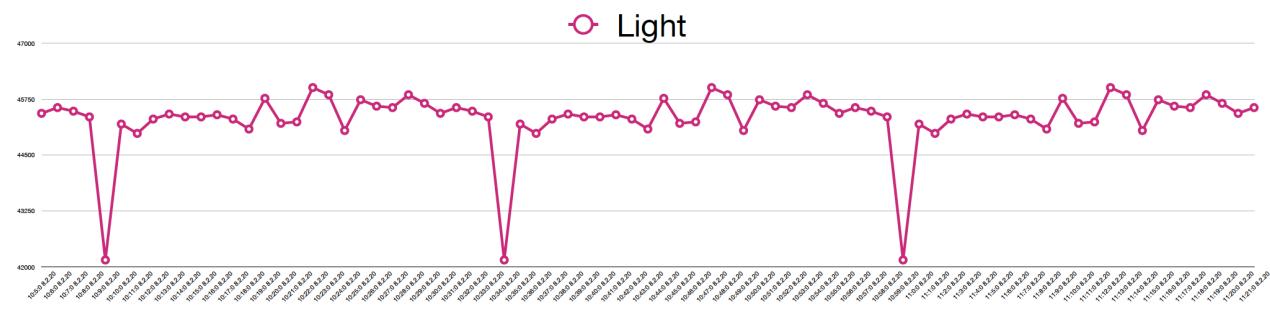


Diagramm Lichtverhältnisse:

Daten von 8.2.20, 10:5:0 bis 8.2.20, 11:21:0



Code Arduino:

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp; // use I2C interface
Adafruit_Sensor *bmp_temp = bmp.getTemperatureSensor();
Adafruit_Sensor *bmp_pressure = bmp.getPressureSensor();

void setup() {
    Serial.begin(9600);
    Serial.println(F("BMP280 Sensor event test"));

    if (!bmp.begin()) {
        Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
        while (1) delay(10);
    }

    /* Default settings from datasheet. */
    bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,    /* Operating Mode. */
                   Adafruit_BMP280::SAMPLING_X2,    /* Temp. oversampling */
                   Adafruit_BMP280::SAMPLING_X16,   /* Pressure oversampling */
                   Adafruit_BMP280::FILTER_X16,    /* Filtering. */
                   Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */

    bmp_temp->printSensorDetails();
}

void loop() {
    sensors_event_t temp_event, pressure_event;
    bmp_temp->getEvent(&temp_event);
    bmp_pressure->getEvent(&pressure_event);

    Serial.print(F("Temperature = "));
    Serial.print(temp_event.temperature);
    Serial.println(" *C");

    Serial.print(F("Pressure = "));
    Serial.print(pressure_event.pressure);
    Serial.println(" hPa");

    Serial.println();
    delay(2000);
}
```

Bewertung Projekt Embedded Systems:

Christian Kolb: 60% (Code, Doku)
Patric Richardson: 40% (Doku, Bauteile)