

Kalibration des Rikamlno-Breakout mit einem Radon-Anreicherungsgefäß

Bernd Laquai, 3.7.2024

Eine große Herausforderung beim Bau eigener Messgeräte ist die Kalibration, so dass am Ende das selbstgebaute Messgerät auch quantitativ brauchbare Messwerte liefert. Für die Kalibration von Messgeräten in der Strahlungsmesstechnik gibt es üblicherweise 2 gängige Ansätze. Der erste Ansatz ist, eine Referenzquelle zu nutzen, deren Strahlungsfeld genau bekannt ist. Dieser Ansatz ist für den Citizen Scientist aber oft nicht so einfach machbar, da Referenzquellen oft nicht kostengünstig verfügbar sind und der Besitz meist auch strahlenschutzrechtlich reglementiert ist. Der andere Ansatz ist die Verwendung eines Referenzmessgeräts, das für die Genauigkeitsansprüche ausreichend genau ist, sowie einer Quelle die für diesen Zweck geeignet und leicht und kostengünstig beschaffbar ist. Letzteres soll hier nun beschrieben werden.

Die Kalibration eines Radonmessgeräts stellt aber noch eine weitere Herausforderung dar. Da die Aktivitätskonzentration des Radon in der Raumluft gemessen werden soll, muss diese in einigermaßen kontrollierter Weise mit Hilfe der Quelle in einem Messvolumen hergestellt werden. Dies kann mit Hilfe eines Anreicherungsgefäßes erreicht werden. Als Quelle eignen sich Stoffe, die natürliches Uran enthalten, welches durch den natürlichen Zerfall ausreichend Radium gebildet hat. Diese Stoffe müssen auch genügend porös sein, damit das Radon das beim Zerfall des Radium entsteht, auch aus der Quelle in ausreichender Form entweichen kann. Das ist bei stark zerkleinertem uranhaltigem Gestein (z.B. Granit aus Flossenbürg) der Fall. Die hier verwendete Quelle besteht aber aus uranhaltigem Düngemittel-Granulat. Es ist bekannt, dass mineralische Phosphat-Düngemittel erheblich mit Uran belastet sind, weil in etlichen Lagerstätten die Rohphosphate auch gleichzeitig stark mit Uran vergesellschaftet sind. Das Uran wird in der Regel nicht abgetrennt, da es in den meisten Ländern keine Grenzwerte für Uran in Düngemitteln gibt. Bei der Herstellung des Düngemittel-Granulats ist es ein Ziel, dass die Pflanzen-Verfügbarkeit des Phosphors besonders hoch ist, dementsprechend ist das Granulat auch hoch-porös. In diesem Fall wurden daher als Quelle 250g Superphosphat 18% der israelischen Firma ICL in Form von kleinen Granulat-Kügelchen verwendet. Das Superphosphat von ICL ist bereits für seinen hohen Urangehalt bekannt. Als Behälter für das Granulat wird eine Standard-Konservenbüchse verwendet. Als Anreicherungsgefäß wurde ein Kunststoff-Eimer mit 10 Liter Volumen verwendet, in dem das Rikamlno-Breakout zusammen mit dem Referenzinstrument, einem RadonEye-Messgerät der Firma FTLab (Ecosense) positioniert wurde.

Das Anreicherungs-Gefäß wurde mit einem einigermaßen dicht schließenden Deckel verschlossen, durch die auch die Kabel geführt wurden. Die Kabelbohrungen wurden mit Klebeband möglichst dicht abgeklebt und zusätzlich mit Knetmasse abgedichtet. Das kleine Messvolumen von nur 10 Liter führt zu einer starken Anreicherung der Luft mit Radon aus der Quelle mit hoher Quellstärke im Vergleich zu einem normalen Wohnraum und den üblichen Eintrittspfaden für das Radon. Dennoch ist der Aufbau der Radon-Aktivitätskonzentration im Messvolumen nicht ganz ideal. Die Dichtigkeit, die erreicht werden kann, ist nicht perfekt, und es kommt an der Kunststoffwand des Eimers zu einem „Plate-Out“, das heißt, die Radon-Zerfallsprodukte stehen nicht vollständig im Gleichgewicht mit dem Radon, sondern heften sich an die elektrisch geladene Kunststoff-Wand des Eimers an. Das führt insgesamt dazu, dass der Aufbau der Aktivitätskonzentration (der „Pile-Up“) nicht ideal mit der Halbwertszeit des Radon

von 3.82 Tagen abläuft sondern deutlich schneller in eine Sättigung übergeht. Dennoch verläuft er nach demselben exponentiellen Gesetz, so dass eine exponentiell sättigende Pile-Up-Kurve sichtbar wird und als Modellfunktion angenommen werden kann. Diese exponentiell sättigende Pile-Up-Kurve wird zur Kalibration von beiden Messgeräten, dem RadonEye und dem RikamIno parallel gemessen. Während der RadonEye die Aktivitätskonzentration in Bq/m^3 misst, liefert der RikamIno die Zahl der Zyklen pro Stunde als Roh-Messwerte. Ziel der Kalibration ist es nun, eine lineare Zuordnung zu finden, mit der man die Roh-Messwerte in Zyklen pro Stunde des RikamIno in Bq/m^3 so umrechnen kann, dass sie möglichst gut mit den Messwerten des RadonEye zusammenpassen. Dies ist mit einer Regressionsrechnung möglich, welche einen Achsenabschnitt und eine Steigung als Kalibrationskoeffizienten für die lineare Umrechnung liefert.



Abb.1: Aufbau der Messung mit dem Anreicherungsgefäß (Eimer), der Düngemittel-Quelle, dem RikamIno-Breakout und dem Referenzmessgerät RadonEye

Der wesentliche Kniff bei der Kalibrationsrechnung mit Excel ist aber, dass zunächst ein exponentielles Sättigungsmodell mit einer Kurvenanpassung in die Messdaten gefittet wird. Für diese Modellkurven-Anpassung wird das Solver Plug-In in MS Excel verwendet. Mit den Parametern für das exponentielle Sättigungsmodell können Werte zu den Messzeitpunkten des RadonEye im Stundentakt durch Interpolation gewonnen werden, die dann koinzident zu den RadonEye Messwerte sind. Somit wird eine Korrelationsrechnung zwischen den koinzidenten bestimmten Werten möglich, welche die Bestimmung der Regressionsgeraden als lineare Zuordnung ermöglicht. Die Regressionsrechnung liefert die Streuung und den Achsenabschnitt der linearen Zuordnung. Diese beiden Werte stellen daher die Kalibrierkoeffizienten dar. Das zugehörige Excel Arbeitsblatt wird mit dieser Beschreibung zur Verfügung gestellt (Download unter <http://opengeiger.de/RikamInoSp250gIndoor.xlsx>) . Ebenso der Arduino-Code, der für die Messung verwendet wurde.

Wichtig ist, dass der Arduino mit dem angefügten-Code nur die Rohdaten auf die Speicherkarte schreibt, also die Perioden-Dauern der Ladezyklen in Sekunden. Alles andere habe ich bisher noch in Form von Post-Processing im Excel gerechnet, insbesondere die Kalibration. Die Kalibrationskoeffizienten lassen sich dann für weitere Auswertungen verwenden. Was gegenüber dem bisher dokumentierten Arduino Code noch hinzugefügt wurde, ist den Code für den HYT221 Temperatur-/Feuchte Sensor. Diese Messwerte lassen sich für eine Temperaturkompensation bei Outdoor-Messungen verwenden, wenn die Temperatur um mehr als 5°C schwankt und ein zusätzlicher Leckstrom durch den Temperatureinfluss in dem RikamIno JFET auftritt.

Auf der Speicherkarte stehen also das Datum (Spalte A im Excel) und die Periode in s (Spalte C) als Messwert (plus Temperatur und Feuchte in Spalte E und F, das sei im Folgenden nur als reine Zusatzinformation betrachtet).

Als erstes wird berechne die Zeit in Tagen (Spalte B) seit Messbeginn berechnet, indem von den Datumswerten der ersten Datumswert abgezogen wird. Dann werden die Zyklen/h in Spalte D berechnet, als die eigentlichen Messwerte, die zur Aktivitätskonzentration proportional sein sollten. Das ist der Kehrwert der Periodendauer in Sekunden multipliziert mit der Zahl der Sekunden pro Stunde.

Als nächstes werden für die Modellparameter in der Modellfunktion der Aktivitätskonzentration:

$$A=A_0*(1-EXP(-LN(2)/T_{12}*t))+B$$

Schätzwerte für A, T₁₂ und B angenommen. Das sind die Werte in Spalte H2:H4. Damit wird eine Kurvenanpassung der Modellkurve an die Messwerte in Spalte D mit dem Namen Fit berechnet. A₀ ist darin der Sättigungswert, T₁₂ die Halbwertszeit (deutlich kleiner als die des Radon wegen der Leckage im Eimer), t die Zeit in Tagen und der Wert B für den Nulleffekt. Diese Kurvenanpassung ist im ersten Ansatz wegen der Schätzung zunächst schlecht. Dann aber wird das Fehlerquadrat der Modellkurve gegenüber den tatsächlichen Messwerten Err in Spalte H berechnet, und der gesamten quadratischen Fehler im Feld H5 aufsummiert. Nun wird das Solver Plug-In in Excel benutzt (siehe Microsoft Excel Dokumentation) um das Feld H5 durch Änderung der Modellparameter in den Feldern H2:H4 zu einem Minimum zu machen. Diese Lösung passt dann in der Regel sehr gut.

Als nächstes werden die RadonEye Messwerte in Spalte M importiert. Die haben nun die Einheit Bq/m³. Da der Radon Eye keine Zeit loggt und man nur weiß, dass die Werte im Stundenraster gemessen sind, werden nun die Zeitwerte des RadonEye in Tagen berechnet (in Spalte L) und zwar mit dem Zeit-Offset tOfst im Feld M8 so, dass der letzte Zeitwert der RadonEye Messung zum letzten Zeitwert der RikamIno-Messung passt, da bekannt ist, dass beide Messgeräte zur selben Zeit ausgeschaltet wurden. Nun kann man mit den Modell-Parametern auch zu den RadonEye Zeiten ebenfalls die Werte der Modellkurve berechnen, so dass die zu den RadonEye-Werten koinzident sind. Diese Werte heißen RikamIno Model (Spalte N). Nun kann man mit den Excel-Funktionen Steigung() und Achsenabschnitt() die Koeffizienten der Regressionsgerade (Felder M2:3) berechnen, die jetzt die Kalibrierkoeffizienten darstellen, mit denen man die Zyklen/h in Bq/m³ umrechnen kann, wenn man davon ausgeht, dass der RadonEye richtig gemessen hat. Mit diesen Werten bekommt man schließlich die kalibrierten Werte der RikamIno-Modellkurve zu den RadonEye Zeiten in Spalte O.

Jetzt kann man aber mit diesen Kalibrierkoeffizienten auch die vom RikamIno zu den Datalog-Zeiten direkt gemessenen Messdaten in Zyklen/h in Bq/m³ umrechnen und in einer Grafik zusammen mit den RadonEye Daten darstellen. Damit kann man nun sehen, wie gut die Kalibration auf den RadonEye geklappt hat.

Dazu sind nun die entsprechenden Grafiken unten gezeigt. Die Zuordnung zu den obigen Schritten ergibt sich durch den Titel der Grafik und die entsprechenden Achsenbeschriftungen. Man kann sich natürlich auch durch Öffnen der Grafik im gezippten Excel die Quelle der Daten anschauen.

Grafiken zu den Kalibrations-Schritten

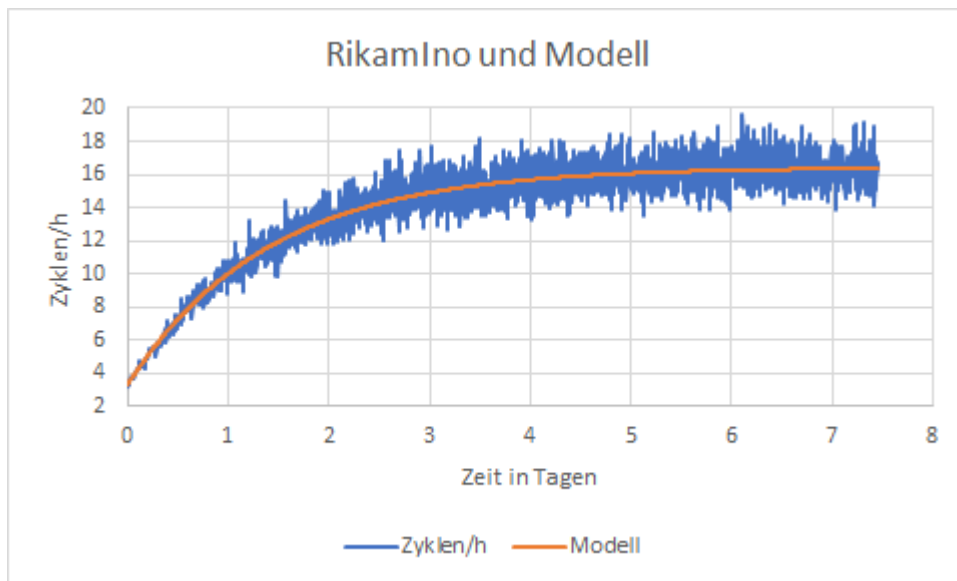


Abb.2: Messwerte des RikamIno Breakouts und der angenommenen und angepassten Modellfunktion für den Pile-Up

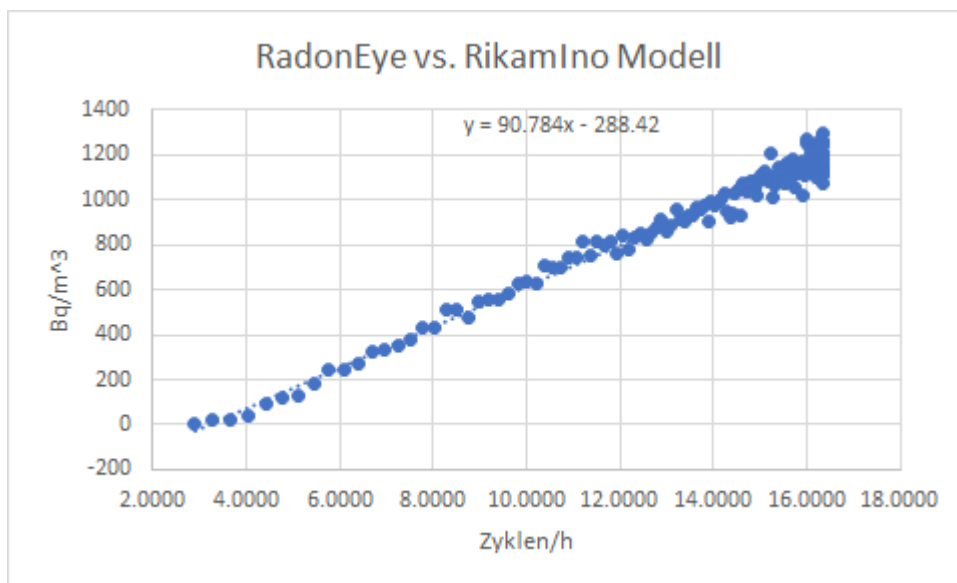


Abb. 3: Korrelation der Messwerte des RikamIno, interpoliert zu den Messzeitpunkten des RadonEye und den Messwerten des RadonEye

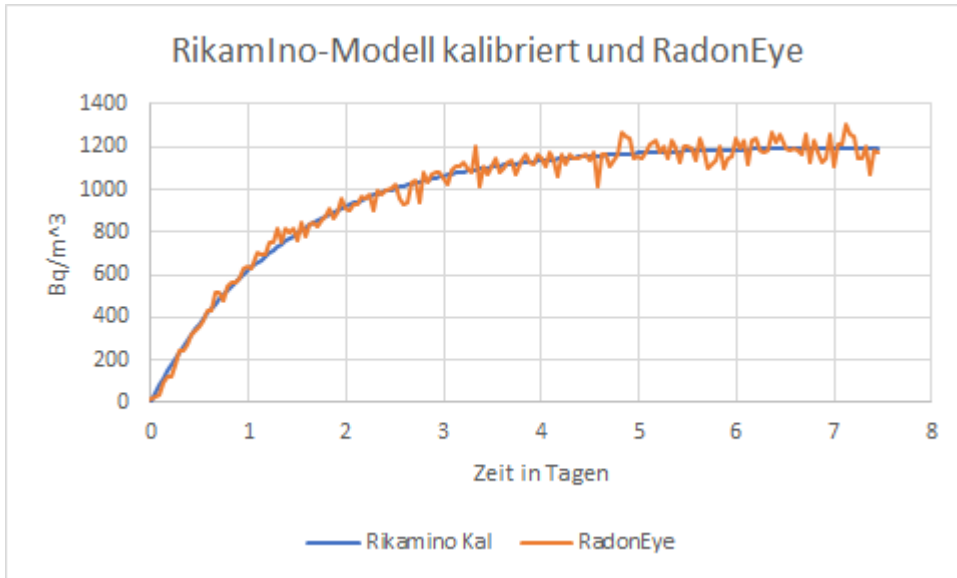


Abb. 4: Kalibrierte Modellfunktion und Messwerte des RadonEye

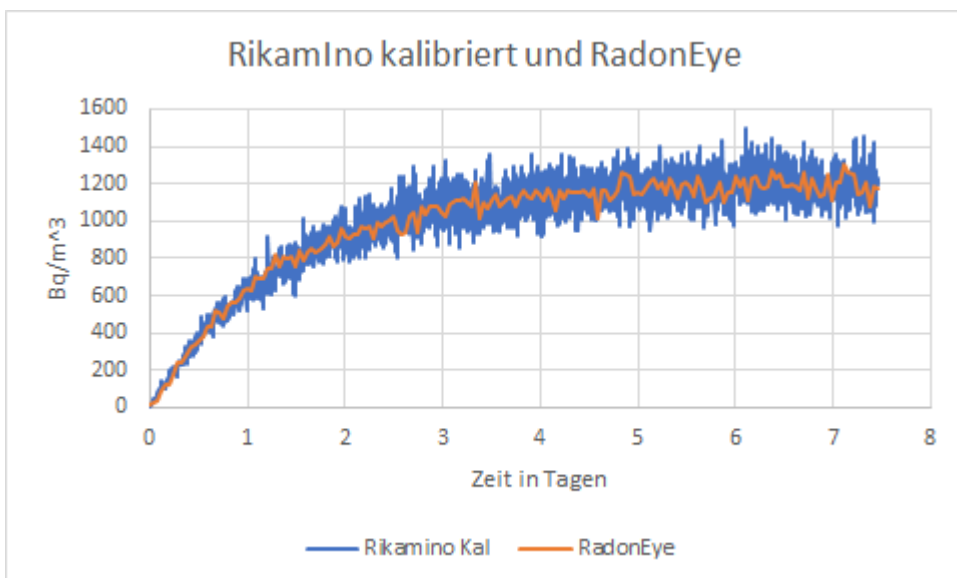


Abb. 5: Kalibrierte Messwerte des RikamIno-Breakout und Messwerte des RikamIno

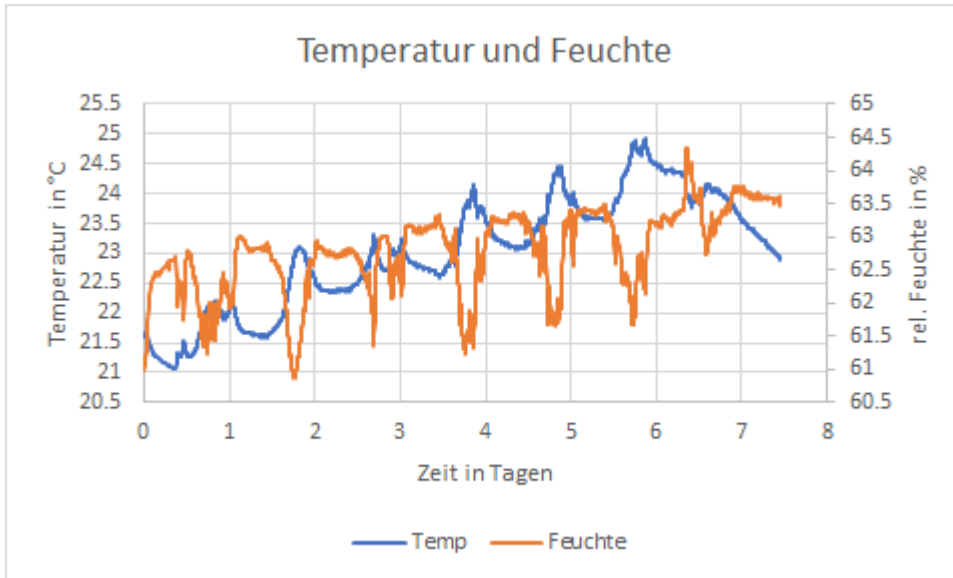


Abb. 6: Temperatur und Feuchte, die vom HYT221 Sensor während der Messung aufgezeichnet wurden, eine Temperatur-Kompensation ist nicht nötig

Arduino-Listing

Hier der verwendete Arduino-Code mit HYT221(HYT939) Temperatur/Feuchte-Sensor:

```
#include <Wire.h>
#define HYT939_ADDR 0x28
#define TFACTOR 99.2909
#define TDELTA 40.0
#define HFACTOR 163.83

#include <SPI.h>
#include <SD.h>
#include "RTClib.h"
#include "rgb_lcd.h"
#define N 50
#define MAXCNT 50

RTC_PCF8523 rtc;
char fileName[15] = "datalog.txt";
File myFile;

void setup()
{
    Wire.begin(); //HYT

    pinMode(10, OUTPUT); // SD Card CS
    Serial.begin(9600);

    if (! rtc.begin()) {
        Serial.println("Couldn't find RTC");
        while (1);
    }
    //if (! rtc.isrunning()) { //DS1307
    if (! rtc.initialized()) {
        Serial.println("RTC is NOT running!");
        // following line sets the RTC to the date & time this sketch was compiled
        // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
        // This line sets the RTC with an explicit date & time, for example to set
        // January 21, 2014 at 3am you would call:
        // rtc.adjust(DateTime(2017, 1, 21, 3, 0, 0));
    }
}
```

```

//rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

if (!SD.begin(10)) {
  Serial.println("SDcard not ready");
}
else
  Serial.println("SDcard ok");

if (!SD.exists(fileName)) {
  myFile = SD.open(fileName, FILE_WRITE);
  myFile.println("###");
  myFile.flush();
}
else {
  myFile = SD.open(fileName, FILE_WRITE);
  myFile.println("-----");
  myFile.flush();
}
Serial.println("Logging started ...");
}

void loop()
{
  //HYT
  unsigned int traw;
  unsigned int hraw;
  double temp;
  double hum;
  int i;
  unsigned char buffer[4];

  //RikamIno
  int out = A0;
  int adcValue = 0;
  unsigned long timeStart, timeStop, dt;
  float volt;

  pinMode(out, OUTPUT);
  digitalWrite(out, LOW); // discharge
  delay(1000);
  digitalWrite(out, HIGH);
  pinMode(out, INPUT);
  delay(1000);
  volt = 0;

  timeStart = millis();
  while (volt < 2.4) {
    adcValue = analogRead(out);
    volt = (float) adcValue*5/1023.0;
    timeStop = millis();
    dt=(timeStop-timeStart)/1000.0;
    delay(1000);
  }

  //read HYT
  Wire.beginTransmission(HYT939_ADDR); // transmit to device #44 (0x2c)
  Wire.endTransmission(); // stop transmitting
  //100ms warten
  delay(100);
  //4 Bytes vom Sensor lesen
  Wire.requestFrom(HYT939_ADDR, 4,true);
  i=0;
  while(Wire.available()) {
    char c = Wire.read(); // receive a byte as character
    buffer[i]=c;
    i++;
  }
  //Rohdaten aus Puffer lesen
  traw=buffer[2]*256+buffer[3];
  hraw=buffer[0]*256+buffer[1];
  //Daten laut Datenblatt maskieren
  traw&=0xffff;
  hraw&=0x3fff;
  traw=traw/4;
  //Rohdaten Umrechnen
  temp=(double) traw/TFACTOR;
  temp=temp-TDELTA;
}

```

```

hum=(double)hraw/HFACTOR;

/*
Serial.print(temp);
Serial.print(' ');
Serial.print(",");
Serial.println(hum);
*/

//logging
DateTime now = rtc.now();

Serial.print(now.day());
Serial.print('.');
Serial.print(now.month());
Serial.print('.');
Serial.print(now.year());
Serial.print(' ');
Serial.print(now.hour());
Serial.print(':');
Serial.print(now.minute());
Serial.print(':');
Serial.print(now.second());
Serial.print(' ');
Serial.print(dt);
Serial.print(' ');
Serial.print(temp);
Serial.print(' ');
Serial.print(hum);
Serial.println();

myFile.print(now.day(), DEC);
myFile.print('.');
myFile.print(now.month(), DEC);
myFile.print('.');
myFile.print(now.year(), DEC);
myFile.print(' ');
myFile.print(now.hour(), DEC);
myFile.print(':');
myFile.print(now.minute(), DEC);
myFile.print(':');
myFile.print(now.second(), DEC);
myFile.print('\t');
myFile.print(dt);
myFile.print('\t');
myFile.print(temp);
myFile.print('\t');
myFile.print(hum);

myFile.println();
myFile.flush();
}

```

Literatur

RikamIno – Ein Radon-Ionisationskammer Break-Out für den Arduino

<http://www.opengeiger.de/RikamIno.pdf>