The Radiation Measurement Shield „Tino“ for the Arduino and Arduino-Compatible Microcontrollers

Bernd Laquai, 20.6.2014

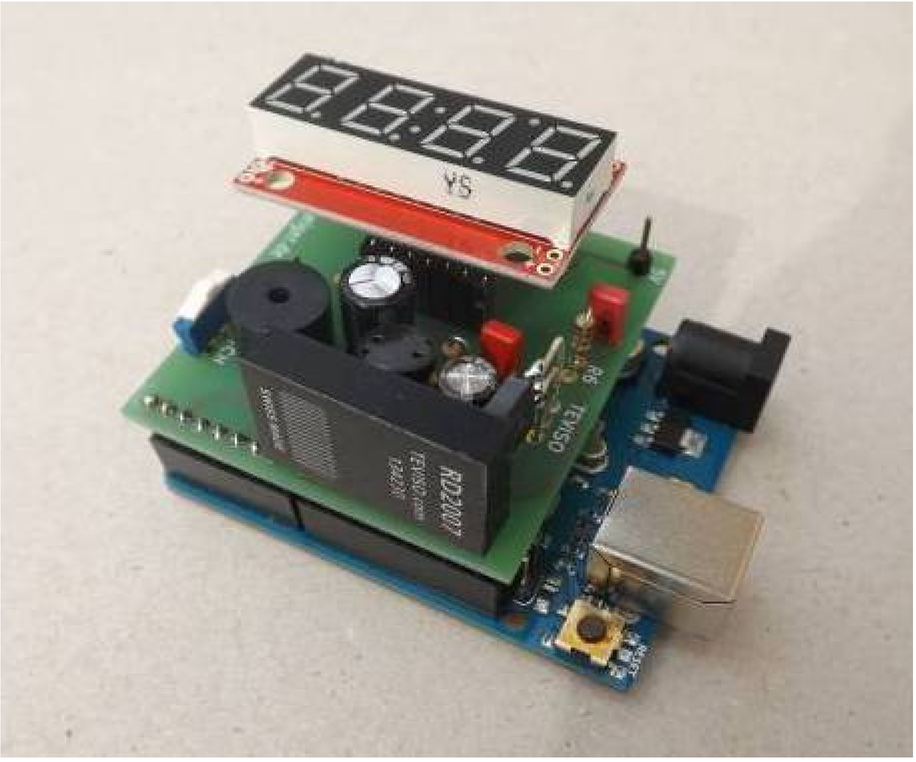

Fig. 1: Completed Tino kit with plugged-in Sparkfun display module on an Arduino UNO

The “Tino” radiation measurement shield

After very positive experiences with the Tino prototype shield as a calibrated measurement instrument for radioactive radiation a professional PCB was developed and manufactured for use with a kit which is now available for the interested community. This PCB and the parts for assembly are designed and selected such that a simple measurement instrument can be created easily and safely with only minimal soldering effort and without much electronic expertise. The basis is the Teviso radiation sensor module, available in different variants as a fully integrated digital PIN diode detector. For the kit a main target was to achieve a low budget solution. Even though, a certain flexibility was provided allowing for example the use of a better display module or a more sensitive sensor module.

Details of the Tino shield concept

Compared to tube based shields for radiation detection the Tino shield offers certain significant advantages. The module can be operated from the 5V supply of the Arduino board and does not need an additional high voltage supply. Even though the high voltage supply for a Geiger-Muller tube is not really powerful, it is still a potential hazard source particularly in education applications, e.g. when the high voltage capacitance accumulates significant amounts of charge at high voltage and gets accidentally touched. In addition the outdoor operation of high voltage powered Geiger-Muller tubes is problematic. Rain and humidity lead to leakage currents on the PCB that may quickly destroy the sensitive input of the counting amplifier or other semiconductors. This problem will not occur with a Teviso module since it is operated from just 5V and uses a regular digital CMOS output. No issue will occur, neither when touching the power supply nor when operated in an humid environment because from 5V supply no relevant leakage currents are to be expected on the PCB that may endanger any sensitive parts. With the small dimensions of the module it was possible to construct a compact shield that does not exceed the dimensions of the Arduino microcontroller board.

The small count rate of the Teviso module (<10cpm) may initially appear as a certain disadvantage compared to a Geiger-Muller tube. Since the statistical standard deviation of a radiation measurement result is proportional to the square root of the number of registered pulses, this automatically means a longer measurement time to reach a similar statistical certainty. For large area radiation sources this is a disadvantage which indeed can only be compensated with more patience. However, with small point-shaped radiation sources, it turns out that a Teviso module may be superior to a Geiger-Muller tube by far. With a Geiger-Muller tube it is for example almost impossible to detect and localize a small uranium mineralization in a piece of ore. The reason is that on a mining waste tip typically a high background radiation is prevailing registered with the full active surface of a counting tube. On the other side, the tube reacts on a small mineralization in the mm range only with a small fraction of its tube surface. In such a situation a tube shows a poor selectivity and a poor spatial resolution. A very small tube does not solve this issue, since with a very small surface the count rate of a small tube also drops proportionally to values even smaller than with a PIN detector.

A similar effect can be seen with watches containing pointers and digits painted on dials with radium containing color. In such a case the Teviso module reacts with a count rate that is high compared to a Geiger-Muller tube since the spatially small area of the sensor gets fully irradiated whereas only a small area of the active tube surface will be irradiated. Also with the small subtleties of an uranium color painting on ancient ceramics a Teviso module shows its high spatial resolution with excellent count rates in a very beneficial way.

A further minor drawback of the Teviso Sensor compared to a Geiger-Muller tube is its sensitivity against electromagnetic radiation from mobile phones, DECT phones and other wireless devices emitted in the range of 800MHz up to several GHz. However, this drawback may eventually become a virtue. It is easily possible that a Tino-Shield generates an alarm when it is working near a mobile phone that gets called and starts transmitting on it's own. However, with a Tino shield it is easily possible to identify the ECO-mode of an DECT-phone and to check if it really stops transmission when it gets placed in the charging cradle. If a Tino shield is brought close to a transmitting DECT phone (less than about 10cm) it will respond with high radiation counts. On the other hand this means that when the intention is to measure radioactivity, a distance of several meters should be kept from mobile phones, WLAN routers and other wireless devices. The most expensive Teviso sensor however, reduces this sensitivity to electromagnetic radiation in the GHz range.

An inestimable advantage of the Teviso sensor is that it comes calibrated with a calibration value for the so-called environmental equivalent dose H*(10) specified by the manufacturer with 15% accuracy (if requested even better). This dose factor (shortly called local dose or when referring to a time interval also named dose rate) is an estimation for the body dose a person would receive when being exposed at this point of the radiation field. This dose is measured in an expanded unidirectional radiation field in a depth of 10cm of a ICRU-phantom. The ICRU-phantom is a sphere of 30cm diameter filled with a material substituting the human soft tissue with a density of 1g/cm^3 and a content of 76.2% oxygen, 11% carbon, 10.2% hydrogen and 2.6% nitrogen. This material is supposed to have similar absorption and scattering properties for ionizing radiation as the human soft tissue.

This specification of a calibration factor means that the user can construct a manufacturer calibrated measurement instrument with minimal effort on his own by just referring to this specification of the sensor manufacturer. Then, such a measurement instrument solution means a significant difference from a tube based device that in fact may just be called a radiation detection device and not a real and reliable measurement instrument due to the missing calibration. Even though a Geiger-Muller tube may be calibrated with simple means (e.g. with potassium chloride) it will react unpredictably on other radiation sources with different radiation energies since only few tube types are compensated to such a wide energy spectrum.

Another important aspect during the design of the Tino shield was the type of display used. The display subject was considered with respect to the readability, the power consumption, the size and the cost of the display module. Basically, with respect to the flexibility for many applications, it is important that the display consumes only a minimal number of digital pins of the Arduino. Therefore a display with a serial interface is beneficial compared to those equipped with the standardized 8 bit parallel interface supported by most of the display controllers. The use of a serial display guarantees that another shield (e.g. Ethernet, WiFi or SD-card) can be plugged on the Arduino in addition before the Tino radiation shield is placed on top of the stack.

The serial display modules differ mainly in the type of technology (LCD or LED displays accordingly). LCD displays are well readable at day light but in the dark most cheap displays miss the background illumination. In contrast they need much less power compared to the LED displays and they can display significantly more characters. However, they are bigger and can hardly be mounted on the shield itself. In contrast serial LED displays normally show just 4 digits with 7 LED segments each and additional decimal points. They are well readable in the dark but when used outdoors at high brightness they often have problems with the contrast and are therefore difficult to read.

Among the LED displays there are two different types available, those operated with constant current (e.g. the one from LiteOn) and those that are multiplexed with a pulsed current such as the Sparkfun display module. Constant current displays need a higher current but generate much less noise in the power supply rails. They are also less expensive than the multiplexed displays. The multiplexed display on the one side are more economic with respect to power consumption but need massive blocking to prevent current spike induced voltage variations in the power supply system that get amplified by the sensor module much like counting pulses. In particular, the Sparkfun display which is multiplexed with about 50Hz may cause such issues when not properly blocked. However, the Sparkfun display is fairly comfortable in operation, even the firmware can be replaced by a self-made firmware version if this is preferred in a specific application. Since all mentioned display alternatives are supported by the Tino shield, special precautions were taken for filtering the power supply rails of the sensor module (LC-filtering).

Generally speaking, it is also possible to operate the shield without any display and to use the PC connected to the Arduino together with a terminal program (serial monitor) for the display of the data.

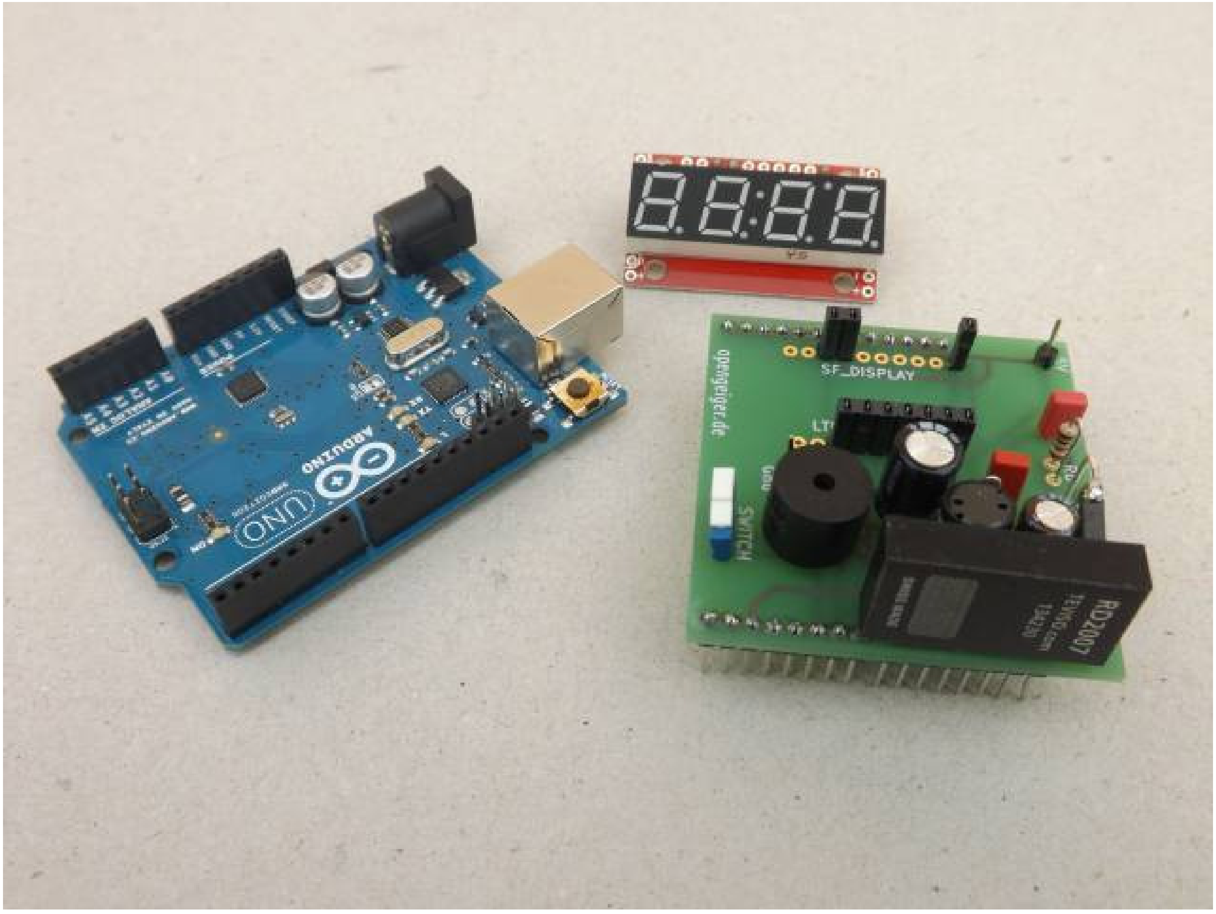

Fig. 2: The different hardware components of the low cost radiation measurement instrument on Arduino basis: the Arduino UNO, the Tino shield and the display module

For an acoustic signaling of the individual disintegration events a piezo sound generator is provided with a sound generation chip integrated in the package. Background is a bad experience with commercial low cost geiger counters. Piezo sound generators typically are housed in a package forming a so called Helmholtz resonator. The manufacturer knows much better at what frequency the generated audio power reaches its maximum and for which frequency the shortest burst (modulated clicks) are possible. In addition, the microcontroller is freed from the unnecessary task of generating sound pulses in the kHz range. And aside of this, the integrated electronic does not get reflected much in the cost of the part. Of course the count events can be easily accessed at the pads provided for the sound generator and can be post processed by a further evaluation unit. Post processing is much easier to do this way compared to the case where the microcontroller generates a sound signal for the piezo. For background monitoring of radiation, the clicking sound is often annoying therefore a switch is provided to turn it off.

As with the prototype the mounting of the Teviso module is designed using perpendicular bended connector wires. This approach has the advantage that the whole shield can be rotated by 90 degrees for a stable positioning on a flat surface while the display still can be read without difficulties. The 3 perpendicular bended connector wires (Gnd, Vcc and Out) can be created without big effort from silver coated copper wire 0.8mm Ø. The soldering pads for it are recessed such that an 1“ array of receptacles still can be placed in between the connector wires and the sensor to have the sensor be changed easily. In a similar way a receptacle array may be used to make the display easily replaceable as well. It was possible to arrange the soldering pads for both displays (LiteOn and Sparkfun) such that both assembly alternatives do not conflict with each other.

The PCB was developed completely for through hole assembly to avoid difficulties with soldering. The two-layered PCB is plated with nickel-gold to avoid oxidation and to ease the soldering. Furthermore the soldering pads were sized quite large (1mm Ø) with drill holes 0.85mm Ø, to ensure the proper fit of the display and Arduino pin arrays.

Using SMD parts a much more dense layout would have been possible of course. But since the area of the shield is determined by the area of the microcontroller board and it is not enlarged by the through hole parts this assembly technology was preferred. Additionally two soldering pads were placed for pins connected to Gnd and to +5V. The reference designators for the parts were placed on the PCB with a silk screen layer to avoid assembly errors. The polarized parts (electrolyte caps and the piezo) are marked with an additional „+“ sign to ensure correct assembly.

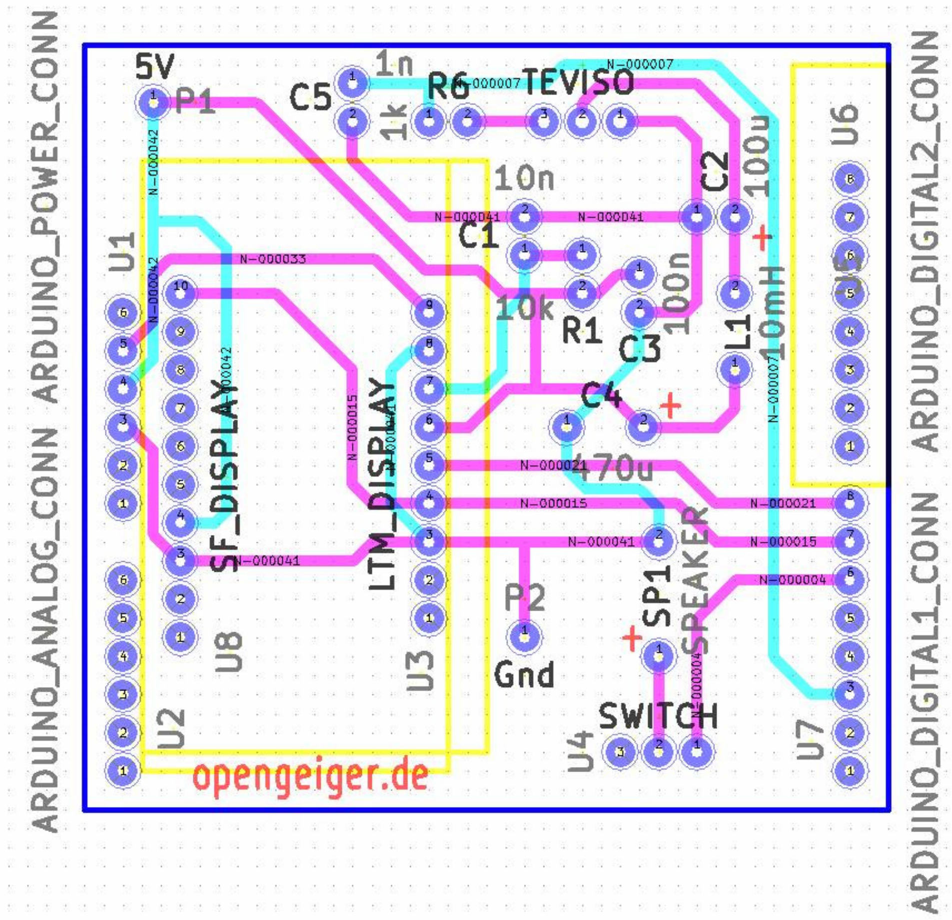

Fig. 3: Layout Plot of the Tino shield PCB

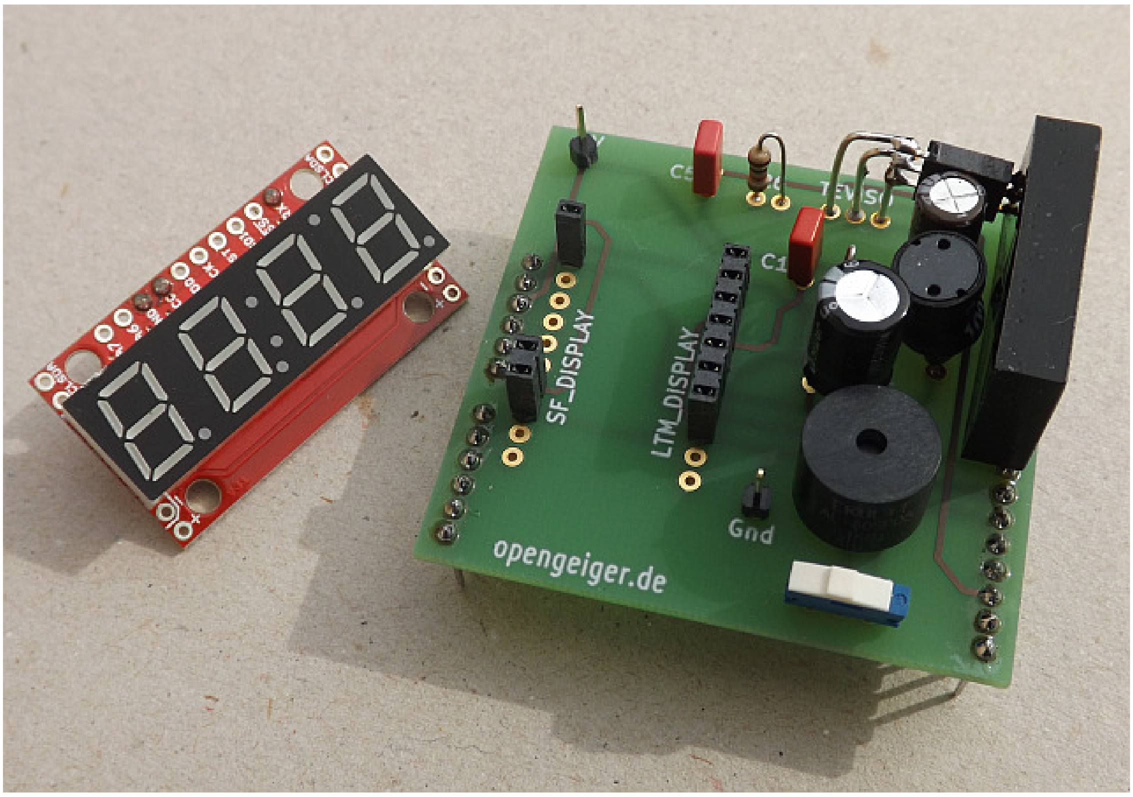

Fig. 4: The assembled shield with removed Sparkfun display

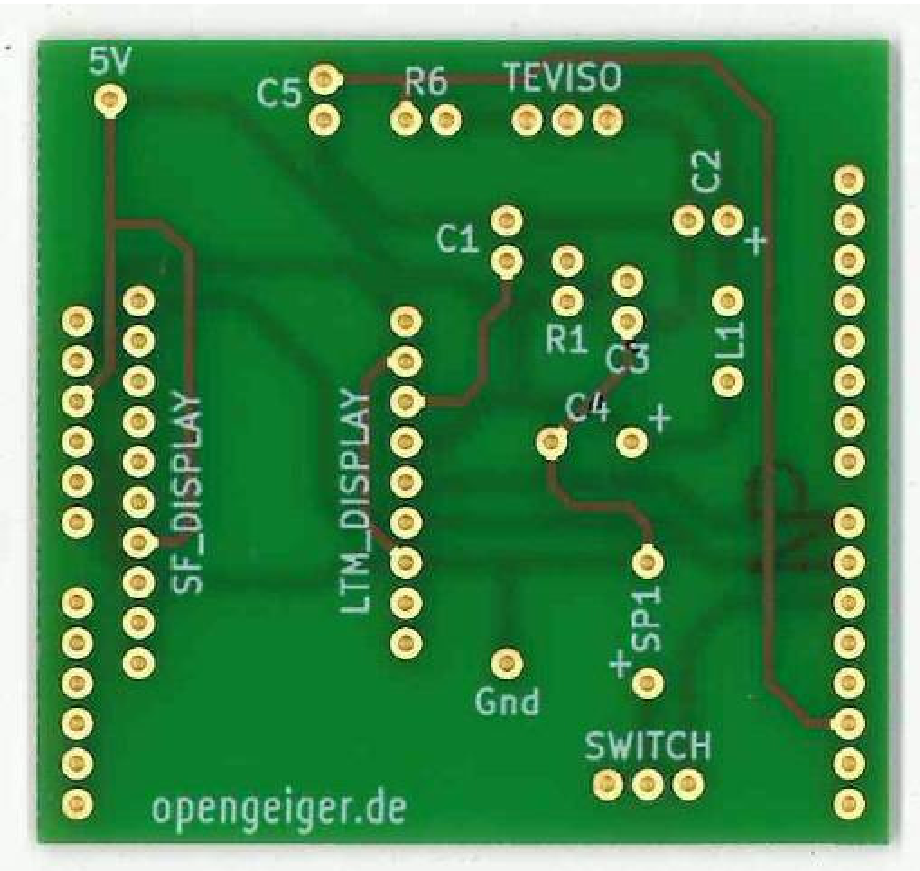

Fig. 5: PCB top layer

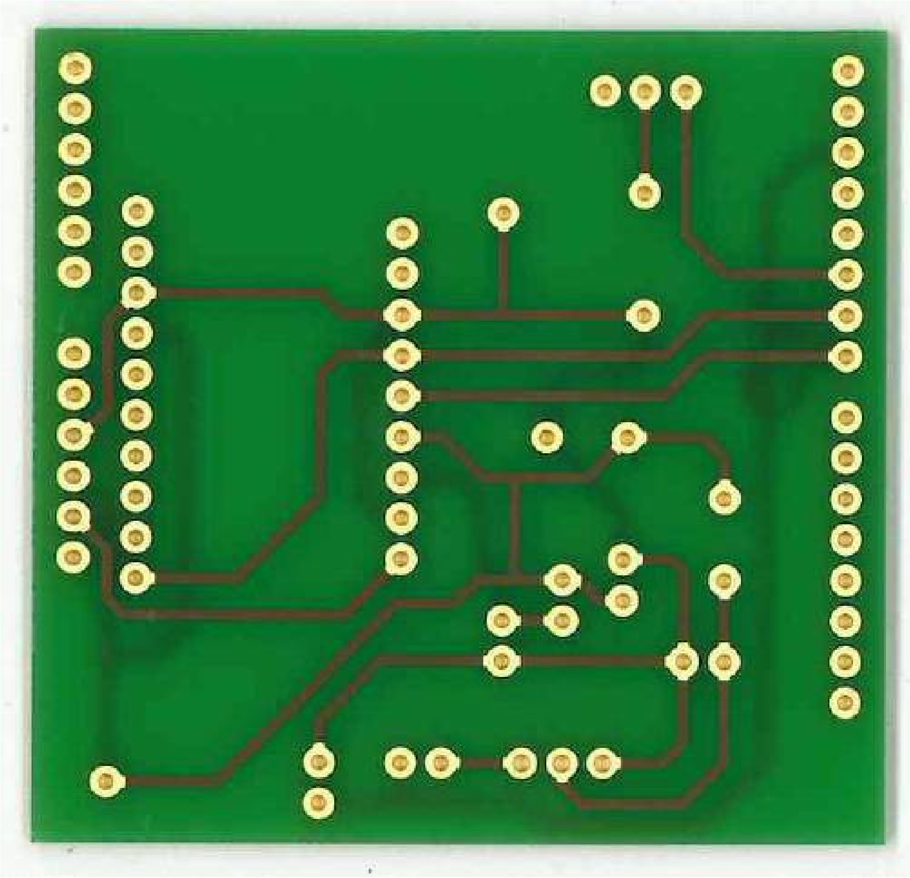

Fig. 6: PCB bottom layer

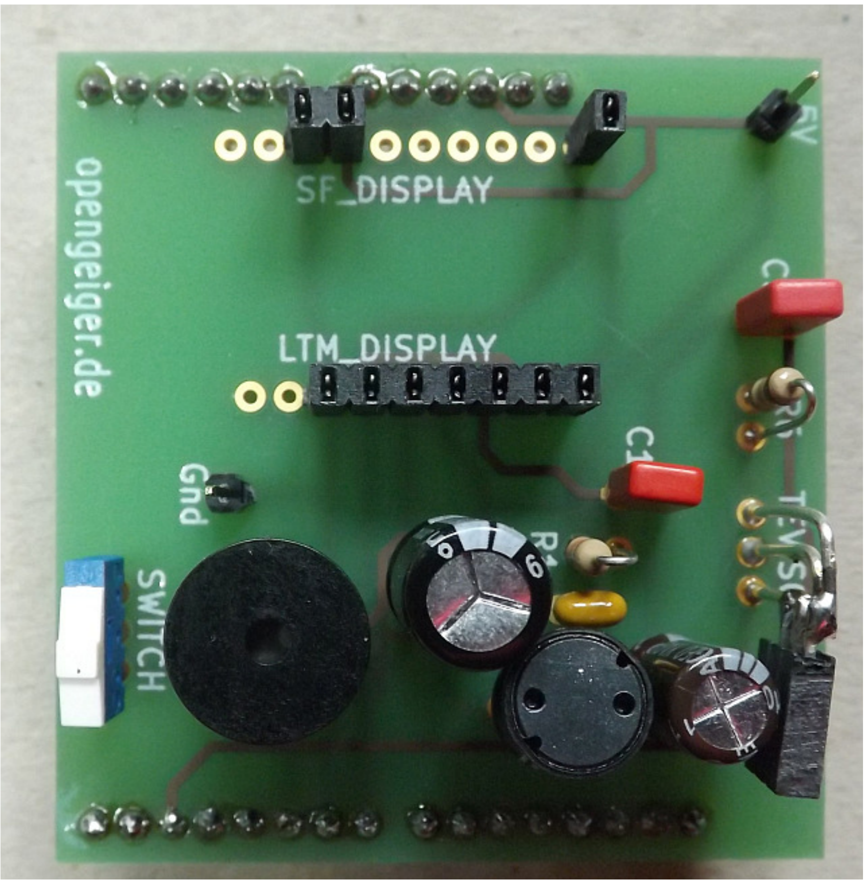

Fig. 7: Assembled Tino shield without Teviso module and without display

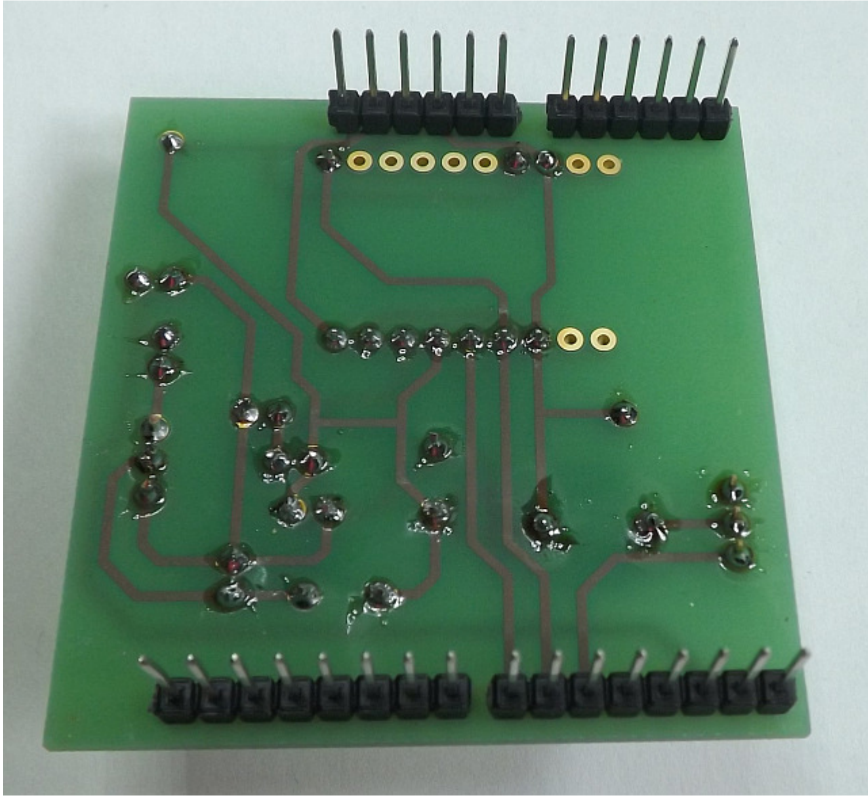

Fig. 8: Assembled Tino shield from the backside

Tino-Shield Bauteile				
Tino Platine		opengeiger		
Piezo Ekulit AL-60 SP 05	Speaker	Conrad	Signalgeber mit Elektronik 85 dB 4 - 7 V/DC	751553 - 62
10mH Induktivität	L1	Reichelt	Stehende Induktivität, 09P, 10mH	09P 10M
470uF/16V	C4	Reichelt	Elektrolytkondensator, 10x12,5mm, RM 5,0mm	RAD 470/16
100uF/10V	C2	Reichelt	Elektrolytkondensator, 105℃, RM 2,5mm	RAD 105 100/35
1nF	C5	Reichelt	WIMA Folienkondensator, Rm 2,5mm 1,0nF	MKS-02 1,0N
10nF	C1	Reichelt	WIMA Folienkondensator, Rm 2,5mm 10nF	MKS-02 10N
100nF	C3	Reichelt	WIMA Folienkondensator, Rm 2,5mm 100nF	MKS-02 100N
1k	R6	Reichelt	Kohleschichtwiderstand 1/4W, 5%, 1,0 K-Ohm	1/4W 1,0K
10k	R1	Reichelt	Kohleschichtwiderstand 1/4W, 5%, 10 K-Ohm	1/4W 10K
Schalter	Switch	Reichelt	Schiebeschalter 1xU, stehend, Print, RM 2,54	SS ESP101
Buchsenleiste 2.54mm		Reichelt	20pol. Buchsenleiste, gerade, RM 2,54, H: 8,5mm	BL 1X20G8 2,54
Stiftleiste 2.54mm		Reichelt	36pol. Stiftleiste, gerade, RM 2,54	SL 1X36G 2,54
LiteOn Display		Digikey	DISPLAY 4DGT ALPHA-NUM RED 0.3"	160-1979-5-ND
RD2007 Sensor		Teviso	RD2007	RD2007
Arduino UNO		Exp-Tech	Arduino UNO SMD R3	EXP-R08-015
Alternativen				
Sparkfun Display		Exp-Tech	7-Segment Serial Display - Grün	EXP-R05-402
RD2014 Sensor		Teviso	RD2014	RD2014

Table 1: Bill of material as currently supplied in Germany

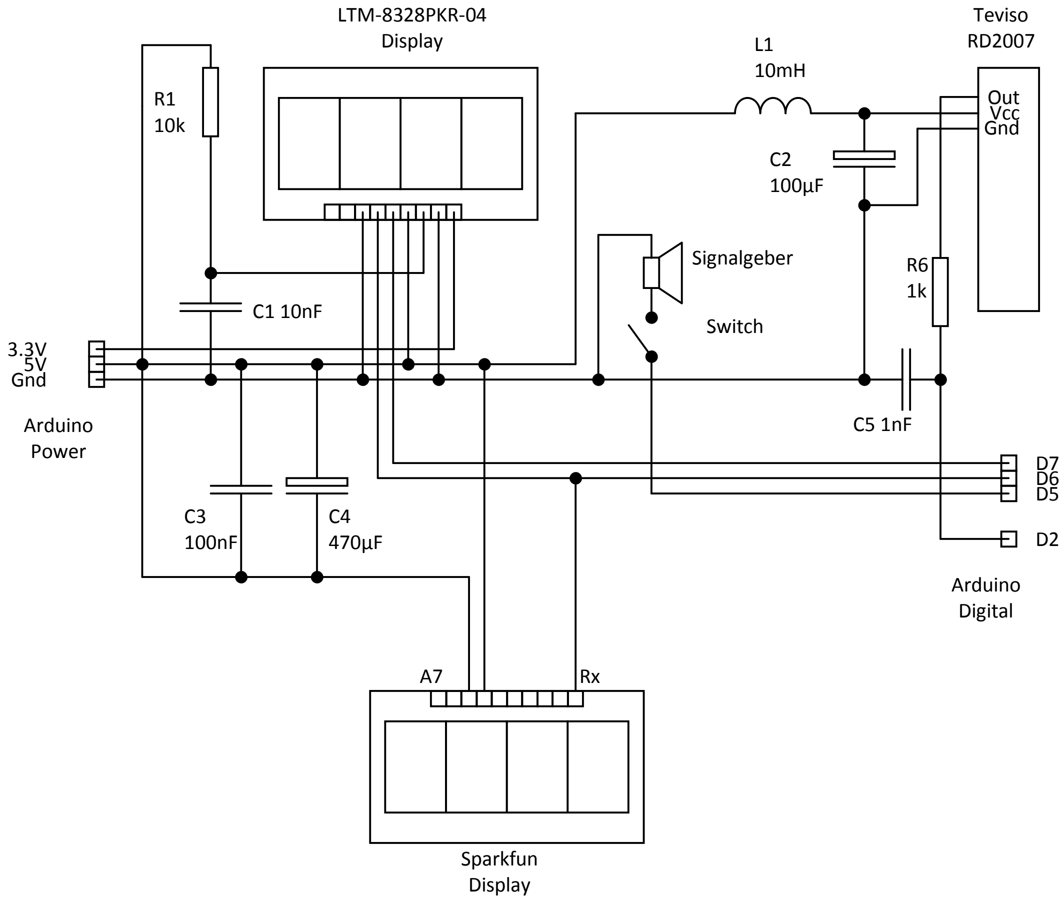

Fig. 9: Circuit diagram of the Tino shield for optional connection of either a serial display from Sparkfun or LiteOn.

Software for the Tino Shield

The software example was further improved compared to the version used with the prototype. In the version used with the LiteOn display the leading zeroes are now suppressed resulting in a significantly reduced power consumption. For both versions (Sparkfun display and LiteOn display) the advice of the Arduino developers was taken into account minimizing the number of instructions in the interrupt service routine by moving most of the processing steps into the main loop.

```
#include <LTM8328PKR04.h>
#define MAXCNT 10
#define CalFactor 3.4

volatile int counter = 0;
unsigned long oldTime = 0;
float rate = 0.0;

const byte dataPin = 6;
const byte clockPin = 7;
int speaker = 5;

LTM8328PKR04 sevSeg(dataPin,clockPin);

void setup()
{
  pinMode(speaker, OUTPUT);
  sevSeg.setLeadingZeros(0);
  int i = (int)(rate*10.0);
  sevSeg.print(i,3);
  attachInterrupt(0, count, RISING);
}

void loop() {
  unsigned long time;
  unsigned long dt;

  time = millis();
  if (counter >= MAXCNT) {
    dt = time-oldTime;
    oldTime = time;
    counter = 0;
    rate = (float)MAXCNT*60.0*1000.0/(float)dt/CalFactor;

    int i = (int)(rate*10.0);
    sevSeg.print(i,3);
  }
}

void count()
{
  counter++;
  digitalWrite(speaker, HIGH);
  delayMicroseconds(50000);
  digitalWrite(speaker, LOW);
}
```

Listing 1: Tino shield software example for the LiteOn display

```
#include <SoftwareSerial.h>

#define MAXCNT 10
#define CalFactor 3.4

SoftwareSerial Serial7Segment(7, 6);
char tempString[10];
volatile int counter = 0;
unsigned long oldTime = 0;
float rate = 0.0;

int speaker = 5;

void setup()
{
  pinMode(speaker, OUTPUT);
  Serial7Segment.begin(9600);
  Serial7Segment.write('v');
  Serial7Segment.write(0x77);
  Serial7Segment.write(0b00000100);
  int i = (int)(rate*10.0);
  char tempString[10];
  sprintf(tempString, "%4d", i);
  Serial7Segment.print(tempString);
  attachInterrupt(0, count, RISING);
}

void loop() {
  unsigned long time;
  unsigned long dt;

  time = millis();
  if (counter >= MAXCNT) {
    dt = time-oldTime;
    oldTime = time;
    counter = 0;
    rate = (float)MAXCNT*60.0*1000.0/(float)dt/CalFactor;

    int i = (int)(rate*10.0);
    char tempString[10];
    sprintf(tempString, "%4d", i);
    Serial7Segment.print(tempString);
  }
}

void count()
{
  counter++;
  digitalWrite(speaker, HIGH);
  delayMicroseconds(50000);
  digitalWrite(speaker, LOW);
}
```

Listing 2: Tino shield software example for the Sparkfun display

The fact that readily usable libraries exist for some popular display modules lead to the selection of the LiteOn display of type LTM-8328PKR-04 (available e.g. from Digikey). For this display a well documented and easy to use library is available written by Doë Goebish. This avoids the writing of many lines of code whereas the library is installed with a few clicks of the mouse. For using the library the downloaded zip-File has to be unpacked, renamed and moved to the standard library folder of the Arduino software (<Path to the Arduino software>/libraries).

For the Sparkfun display software support is also provided with many examples. For this display however, it is not required to download a specific library. Instead the regular Arduino software serial library can be used to generate a RS232 interface on two pins. Only the Tx pin is really required which has to be connected with the Rx pin of the display. For the communication with the Sparkfun display the serial interface was used because it is easier to handle even though the I2C or SPI interface are also supported by this display.

In the example program the initialization of the serial communication is achieved in the setup() routine by simply creating an object „Serial7Segment“ with the software serial method such that the digital pin 7 is used as Rx pin and pin 6 acts as Tx on the Arduino side. Actually pin 7 is unnecessary since only writing to the display occurs and the display is never read. The way how it is written to the display or what may be send as specific initialization is described in detail with many examples in the GitHub repository. To send the right format for the data, the measurement value (dose rate in µSv/h) is first converted with a sprintf() statement to a string of 4 characters temporarily stored in the variable tempString. It is then written to the serial7segment object with the .print() method. The approach for the LiteOn display is similar but due to the smaller flexibility it requires less code.

Aside of the standard routines setup() and loop() only the count() routine is added to service an interrupt when counting pulse appear. In the interrupt service routine (ISR) named "count()" only the actual counter variable is incremented to ensure that other interrupt controlled functions of the operating system are not blocked unnecessarily. The calculation of the dose rate is done in the main loop() executed periodically even when no counting pulses get registered. The routine count() only gets triggered when a rising edge is recognized by the interrupt logic at the interrupt input 0 (digital pin 2). This behavior is defined with the statement „attachInterrupt(0, count, RISING);“ . With other Arduino boards than the UNO (e.g. the Due) this may eventually be different. In the normal case the microcontroller is cycling in the loop() routine, calculates the actual dose rate and outputs it on the display. As soon as the interrupt condition on pin 2 is met, it saves all work and executes what is defined in the ISR named count(). In the ISR the counter variable is incremented and the signal generator is turned on for 50ms to produce an audible click. After this few instructions program execution returns to the main loop() routine. The variable "counter" must be declared as volatile since it may change in the ISR and the compiler must be instructed to reload the value from main memory each time the program accesses it without applying any optimization for avoiding this.

A special aspect of this counting routine is the fact that unlike with other programs it does not count the number of pulses in a given interval, it rather measures the time needed to register a specified number of MAXCNT pulses in order to determine the rate. This approach proved to be advantageous since for some samples a long time may be required to register a statistically representative number of pulses. For other samples many pulses are registered very quickly. When it is required to react on a change of rate e.g. from a small rate to a large rate it would be disadvantageous to use a large time interval also for the high rate. In such a case the increase to the high rate would be noticed not before the long time interval chosen for low dose rate samples is over. With the used approach however, the statistics (the variance of the measurement result) stays constant independent of the counting rate. Since the Teviso sensor may need almost half an hour for some natural low dose rate samples to register 10 pulses, this strategy preserves a further half of an hour of measurement time because its sees the increase much faster.

In the loop() routine the rate will be re-calculated only for the case the maximum number of pulses MAXCNT is reached. In this case the variable „rate“ will be updated. The time difference dt which has passed since the last time MAXCNT was reached and the actual time is measured to calculate the rate. The rate and the calibration factor will finally be used to calculate the dose rate. As long as this calibration factor is not set to just 1 but in case of the RD2007 sensor to 3.4, the result is the dose rate in µSv/h and not the simple count rate of pulses per time.

For writing the dose rate to the display e.g. to the LiteOn display an object (a variable) sevSeg of type LTM8328PKR04 has to be created. The use of this object is a requirement of the specific display library. With this object declaration the pin numbers for the data and the clock pin are passed over to the library methods (routines). With this object declaration the methods become available and can be used to access the display. The .print() method has to be used to the display. The actual value is passed over with the integer variable i followed by a byte variable that defines the position of the decimal point. Since the rate is a floating point number it is calculated with floating point arithmetic. Therefore the output on the display requires a conversion to the integer format with the consequence of truncating the fractional part of the number. When the dose rate is to be output in µSv/h, at least one digit is required after the decimal point. In order not to loose it with the integer truncation, the rate value has to be multiplied with 10. This gets compensated again by shifting the decimal point behind the 3rd digit.

The correct operation of the program can easily be verified with a pulse generator. If it is adjusted to output periodical pulses with a pulse width of about 200µs at a frequency of 5.6Hz then 10 pulses (MAXCNT=10) will require 10*1/5.6Hz = 1.786 seconds to be registered by the microcontroller. The Arduino measures this time in milliseconds, so this are 1790ms. The program therefore calculates that the dose rate is 10*60*1000/1786/3.4 or nearly 100µSv/h. This value can be easily remembered as a quick and simple check.

After powering up the Arduino the display needs a couple of seconds to initialize. After initialization it shows “.0” as it is written to the display during the setup() routine. The Tino shield measures the normal environmental radioactivity with about 0.1µSv/h which gets displayed after about 30min (the RD2007 assumed). Close to a container with 100g potassium chloride (KCl) 99.9% from the pharmacy it shows the value of “.4”, meaning 0.4µSv/h. When using a small value for MAXCNT e.g. MAXCNT=10, it needs to be taken into account that according to statistical theory the standard deviation of the dose rate D is then about D/√(MAXCNT) or approx. D/3. The variation of the value is therefore not really negligible. A decrease of the standard deviation it is only possible by increasing MAXCNT accordingly compromising the measurement time. In front of an old wrist watch with a radium painted dial the display can easily show numbers of above “50.0” meaning dose rates of above 50µSv/h due to the complete irradiation of the small PIN-diode in the sensor and with a strong radium source in immediate vicinity. This in contrast takes just 3.5s when using a value of MAXCNT=10. It shows that it is important to correctly adjust the MAXCNT parameter to the expected spread of the radiation sources to optimize measurement time and accuracy.

The example programs can be easily modified upon need to any desired application. Just as an idea for modification the signal generator could generate a constant alarm when the dose rate crosses a certain threshold. Further improvements and extensions are quickly done due to the straight forward concept and its simplicity of implementation.

Photos of applications „Tino“ radiation measurement shield with the LiteOn Display

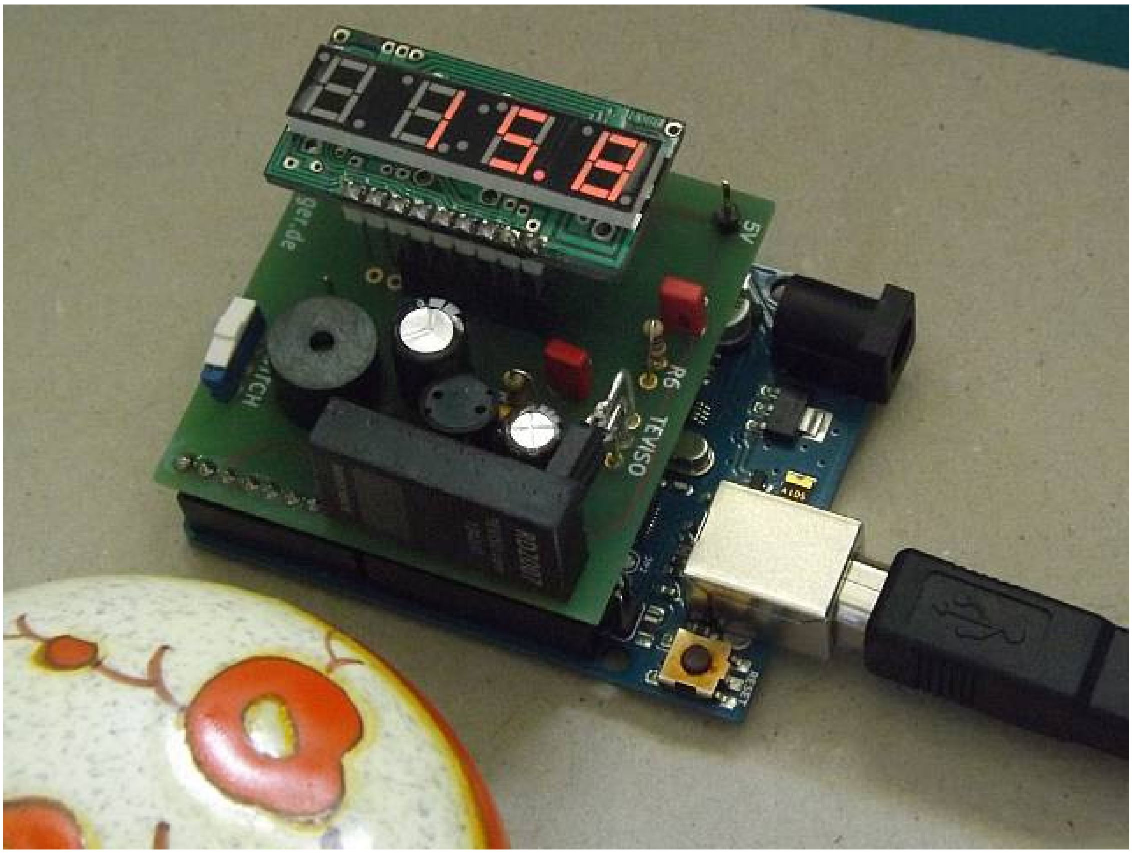

Fig. 10: Tino in front of an old ceramic pill container painted with uranium colors

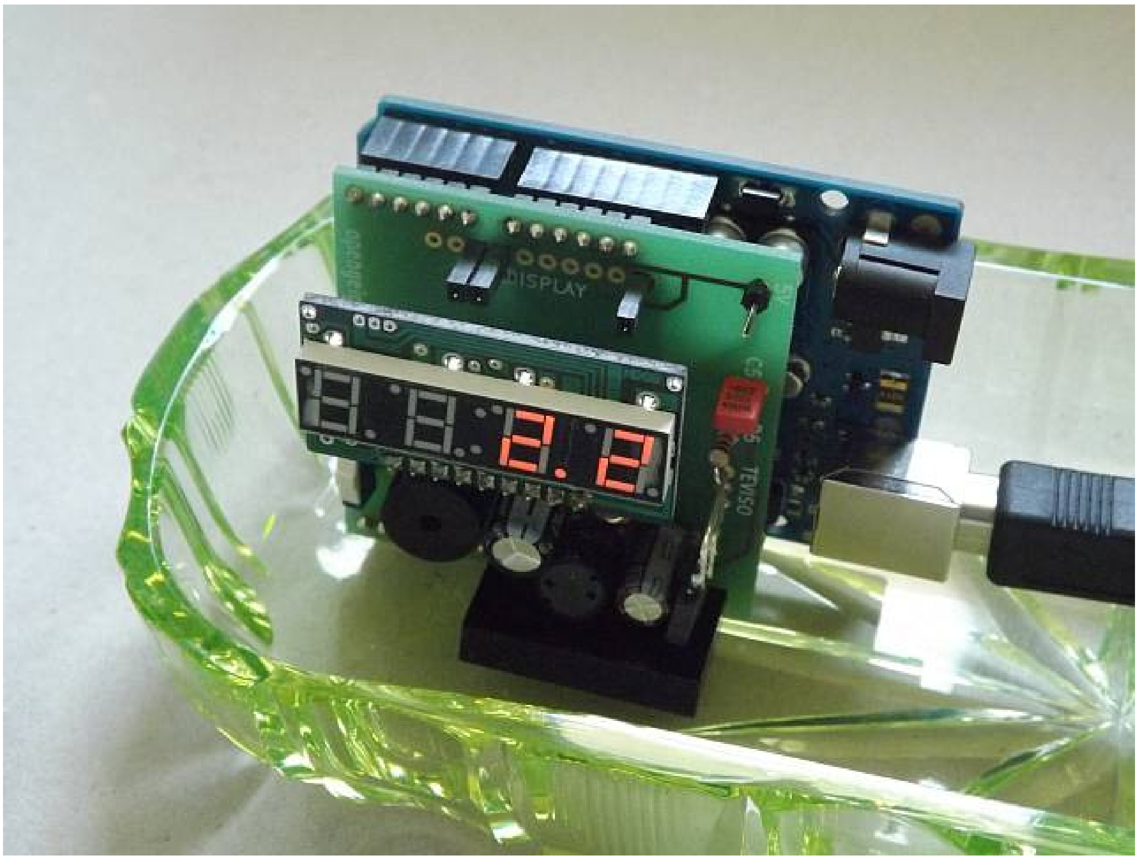

Fig. 11: Measurement of a glass bowl colored with „Anna-Gelb“ an uranium glass color

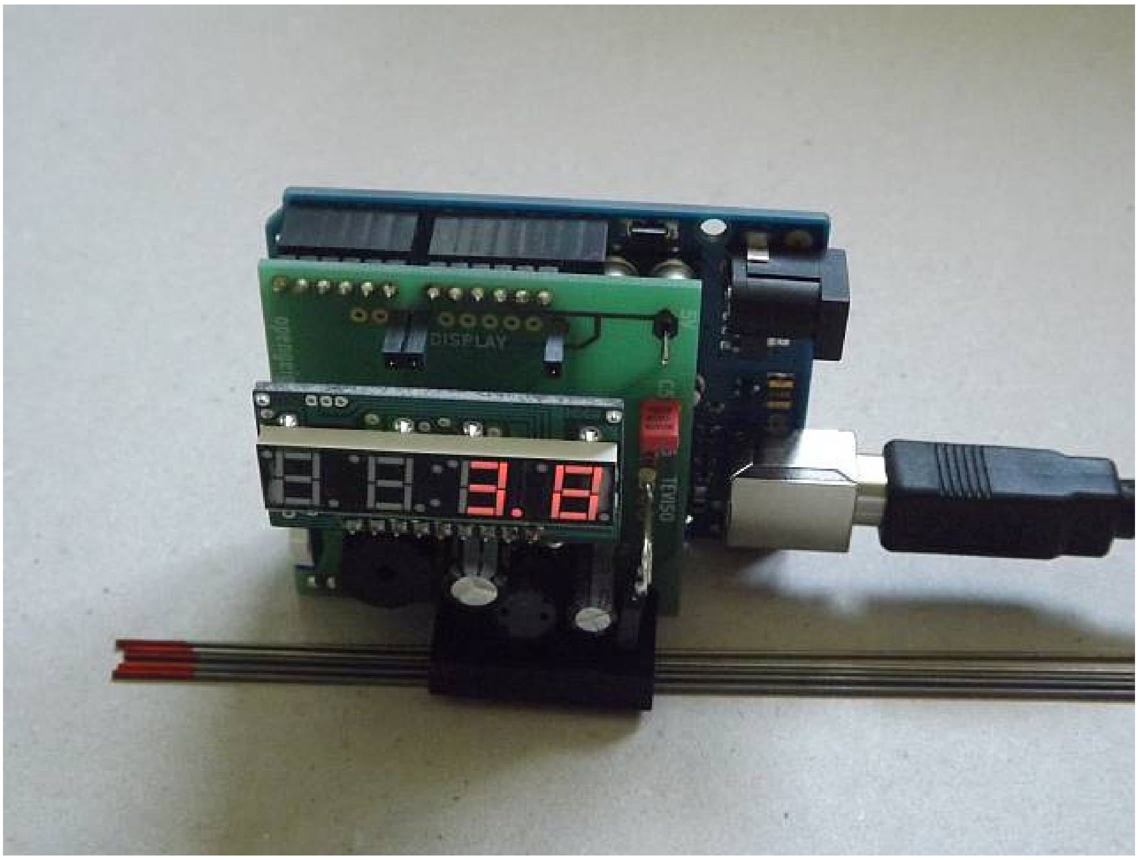

Fig. 12: Tino measuring WT-20 welding electrodes (Thorium oxide content 1.8-2.2%)

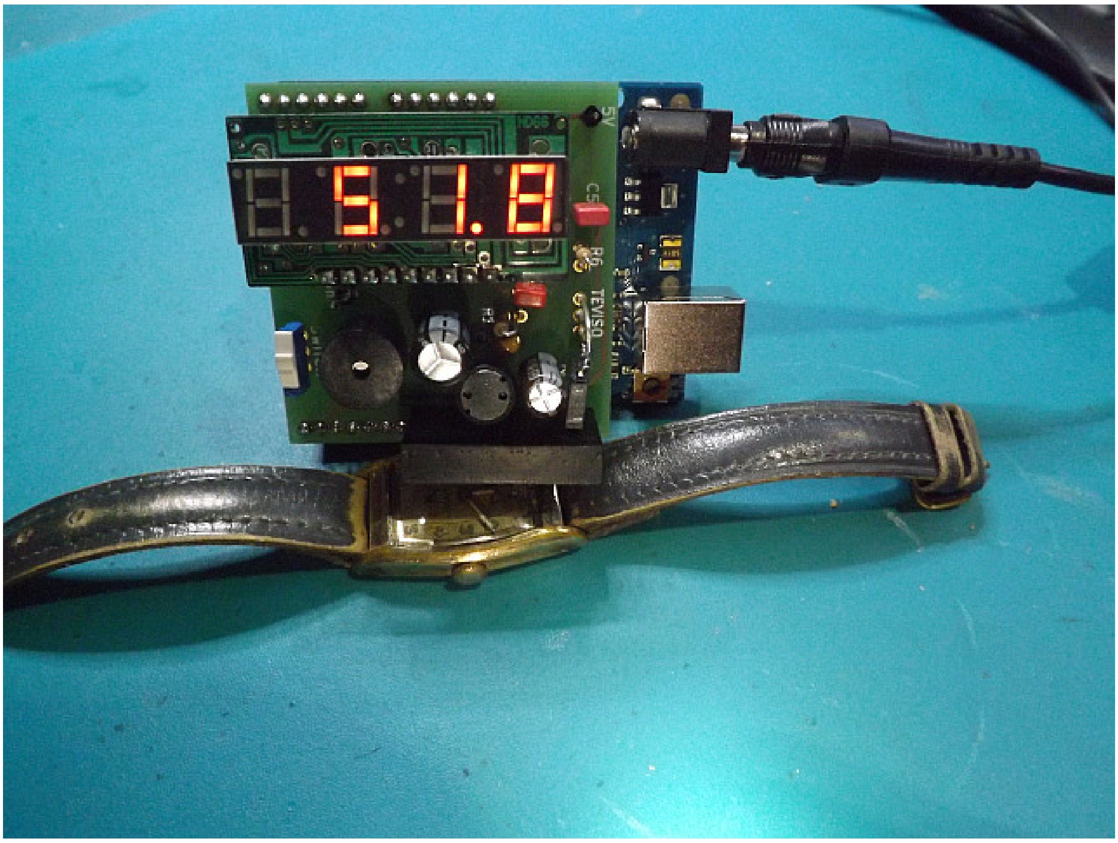

Fig. 13: Measurement of an old wrist watch with radium containing pointers

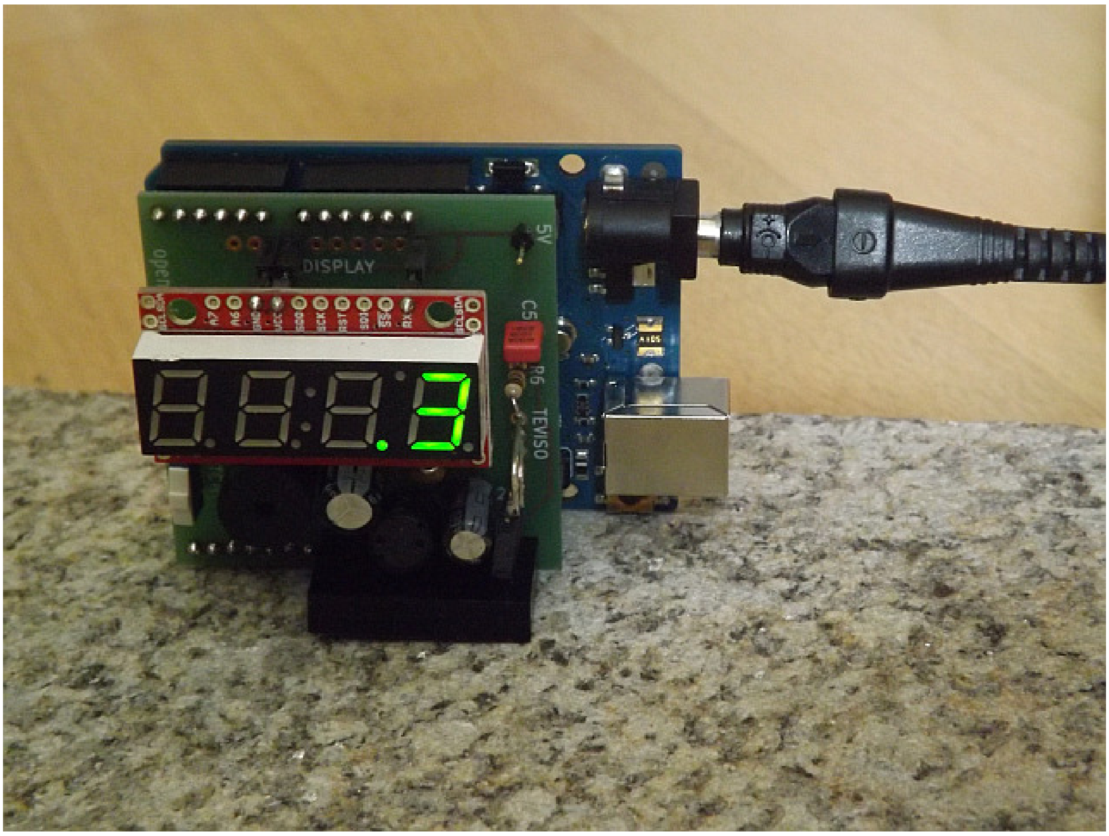

Fig. 14: Tino Shield (with Sparkfun display) placed on a granite tile from Flossenbürg, measurement time 10min.

Links to the part suppliers (Germany) and software libraries

Teviso (online-Shop see „Contact“)
http://www.teviso.com

Reichelt Elektronik
http://www.reichelt.de

Conrad Electronic
http://www.conrad.de

Digikey
http://www.digikey.de

Exp-Tech
http://www.exp-tech.de

Lite-On LTM-8328PKR-04 4x7 LED segments display library for Arduino
https://github.com/goebish/LTM8328PKR04/

Datasheet des LTM-8328PKR-04 from Digikey:
http://media.digikey.com/pdf/Data%20Sheets/Lite-On%20PDFs/LTM-8328PKR-04.pdf

Sparkfun display (green)
https://www.sparkfun.com/products/11440

Ekulit signal generator with internal control, type: AL - 60 SP 05
http://www.ekulit.de/fileadmin/Bilder/produkte/signalgeber_mit/170050%20AL-60SP05.pdf