

Strahlungswächter

Radioaktivität messen, aufzeichnen und in das Smart Home integrieren

Spätestens seit dem Reaktorunglück von Tschernobyl und der Nuklearkatastrophe von Fukushima ist radioaktive Strahlung ein wichtiges Thema. Durch die alternden Atomkraftwerke in Europa und den Krieg in der Ukraine mit dem Beschuss des größten Atomkraftwerks Europas in Saporischschja ist das Thema allgegenwärtig. Das CEM Radioaktivitätsmessgerät DT-9501 kann Alpha-, Beta-, Gamma- und Röntgenstrahlung nachweisen und verfügt über ein großes LC-Display, eine Alarmfunktion sowie eine Bluetooth-Schnittstelle zur Übertragung der Messwerte. Nach einem kurzen Test und der Funktionsbeschreibung wird aufgezeigt, wie man mit einem Mikrocontroller wie dem ESP32 die Daten über einen Webserver zur Verfügung stellen kann, um diese beispielsweise in die eigene Heimautomatisierung einzubinden.



Leserwettbewerb



*siehe Seite 112

Dr. Peter Tschulik

hat für seinen Beitrag zum Leserwettbewerb einen Gutscheincode* über 200,- Euro erhalten!

Grundlagen

Die uns umgebende Strahlung aus dem Erdboden, der Luft und dem Welt- raum ist so gering, dass sie dem Menschen nicht schadet. Ganz anders schaut es mit der sogenannten radioaktiven Strahlung aus, die von schad- haften Kernkraftwerken oder Nuklearwaffen ausgeht, die zu schweren Schäden wie Hautverbrennungen, Erbgutveränderungen oder Krebs füh- ren können. Genau genommen ist der Begriff „radioaktive Strahlung“ nicht richtig, da die Radioaktivität vom emittierenden Stoff ausgeht. Leider ist diese radioaktive Strahlung für uns Menschen unsichtbar. Sie kann grob in die drei Strahlungsarten Alpha-, Beta- und Gammastrahlung unterteilt werden (siehe Übersicht über die drei Strahlungsarten).

Zum Glück gibt es Messgeräte, die in der Regel als Geigerzähler be- zeichnet werden und diese Strahlung messen können. Billige Geräte zei- gen nur die radioaktiven Zerfallsprozesse durch ein Ticken an. Gängige Messgeräte messen vor allem Beta- und Gammastrahlen über Geiger- Müller-Zählrohre. Die offiziellen international genormten Maßeinheiten für die Strahlung sind Gray (Gy) und Sievert (Sv) für absorbierte und äqui- valente Dosen. Die gängigste Einheit ist Mikrosievert pro Stunde ($\mu\text{Sv/h}$). Die Umgebungsstrahlung ist normalerweise geringer als $0,10 \mu\text{Sv/h}$.

Übersicht über die drei Strahlungsarten der Radioaktivität

Alphastrahlen:

Die Teilchen der Alphastrahlen sind groß, weswegen sie schon von Papier und anderen Materialien abgeschirmt werden. Sie können zwar im Normalfall die menschliche Haut nicht durchdringen, allerdings eingeatmet, geschluckt oder über Wunden in den Körper gelangen. Außerdem beschränkt sich ihre Reichweite in der Luft auf wenige Zentimeter. Fälschlicherweise werden sie daher oft als ungefährlich betrachtet. Dies stimmt beispielsweise nicht für gasförmige Alphastrahlen wie das oft aus dem Erdboden in den Keller einströmende Radon. Eingeatmet befindet es sich dann in unmittelbarem Kontakt mit dem Lungengewebe. Dort kann es dann beispielweise Krebs auslösen. Viele billige Geigerzähler können Alphastrahlung nicht erkennen, da sie von den Zählrohren abgeschirmt werden.

Beispiel für Alphastrahler: Radium, Radon, Uran und Thorium

Betastrahlen:

Diese Strahlungsform besteht aus sich sehr schnell bewegenden Elektronen bzw. deren Antimaterie-Pendants Positronen, die sehr klein sind und nur eine geringe Reichweite haben. Deshalb werden sie nur durch Bleiplatten oder ähnliche Stoffe abgeschirmt, wobei auch Kleidung eine gewisse Abschirmwirkung aufweist. Betastrahlen können in jene Schicht der menschlichen Haut eindringen, die neue

Hautzellen produziert, dort länger verbleiben und zu einer schweren Schädigung des Erbguts und zu Krebs führen.

Beispiel für Betastrahler: Strontium-90, Carbon-14, Tritium und Schwefel-35.

Gamma- und Röntgenstrahlen:

Diese Strahlen sind elektromagnetische Wellen ähnlich dem Licht und bestehen aus Photonen. Gegenüber dem Licht haben sie allerdings eine erheblich kürzere Wellenlänge und sind energiereicher. Die gegenüber Alpha- und Betateilchen fehlende Ladung bewirkt, dass Gammastrahlen schwer abschirmbar sind. In biologisches Gewebe können sie tief eindringen und zerstören dort chemische Bindungen und große Moleküle unserer DNA. Röntgenstrahlen sind Gammastrahlen sehr ähnlich und dringen ebenfalls tief in den menschlichen Körper ein, was man sich in der Medizin zunutze macht. Hohe Dosen von Gammastrahlen führen zur Bildung von Krebs oder dazu, dass sich betroffene Zellen nicht mehr teilen können und somit absterben. Dies wird im Volksmund als Strahlenkrankheit bezeichnet und hat in Tschernobyl bei den Hilfs- und Rettungskräften nach wenigen Tagen zum Tod geführt. Kleidung schützt nicht vor einer Kontamination.

Beispiel für Gammastrahler: Iodine-131, Caesium-137, Cobalt-60, Radium-226 und Technetium-99m.

Eine andere Möglichkeit ist die Zählung der eintreffenden Impulse, „Counts per Minute“ (CPM). Weitere aussagekräftige Informationen über die zulässige Strahlendosis findet man z. B. unter [1]. Ein Geiger-Müller-Zählrohr ist mit einem Niederdruck-Gasgemisch gefüllt. Wenn es mit einer entsprechenden Spannung vorgespannt wird (typischerweise mit 400-500 V), ionisiert das Gas und leitet kurzfristig Strom, sobald es von Strahlung getroffen wird. Je nach Rohrtyp ist es möglich, Alpha-, Beta- und Gammastrahlen nachzuweisen.

Aus aktuellem Anlass kaufte ich das Radioaktivitätsmessgerät CEM DT-9501 (Bild 1) im ELVshop [2], dessen Daten sich vielversprechend anhörten und das für einen universellen Einsatz in der Umwelttechnik, Entsorgungsbereichen, Industrie, Medizintechnik, bei Sicherheitsorganen und im privaten Bereich z. B. bei Radonmessungen oder an Baumaterialien vorgesehen ist. Das kompakte Handmessgerät mit einem großen beleuchteten LC-Display mit digitaler Anzeige und einer integrierten Uhr für Uhrzeit und Datum bietet

- die Erfassung und qualitative/quantitative Auswertung von Alpha-, Beta-, Gamma- und Röntgenstrahlungswerten
- drei Strahlungsmessmodi für Alpha-, Beta- und Gammastrahlung oder nur Beta- und Gammastrahlung oder nur Gammastrahlung und eine automatische Messbereichswahl
- einstellbare Grenzwerte mit Alarmfunktion
- die Umrechnung in verschiedene Maßeinheiten
- einen Datenspeicher für bis zu 4000 Messwerte
- eine Bluetooth-Schnittstelle und PC-Auswertesoftware

Unboxing

Das Messgerät kommt in einem Kunststoffkoffer, der mit weichem Schaumstoff ausgekleidet ist. Erfreulicherweise werden auch gleich vier Mignon-Batterien für den Betrieb mitgeliefert, sodass eine Inbetriebnahme schnell erfolgen kann. Ebenfalls im Koffer befindet sich der Griff für den Kunststoffkoffer, der einfach in die beiden Löcher eingesteckt wird. Außerdem liegen eine englisch- und deutschsprachige

Anleitung sowie die CD mit der PC-Software bei. Die übersetzte deutsche Bedienungsanleitung könnte meiner Meinung nach deutlich besser sein. Deshalb auch nachfolgend eine kurze Beschreibung der Inbetriebnahme und Bedienung. Das Gerät selbst macht einen sehr robusten Eindruck, liegt gut in der Hand und verfügt über eine seitliche Gummibeschichtung als zusätzlichen Schutz.



Bild 1: CEM Radioaktivitätsmessgerät DT-9501

Nach dem Lösen der Schraube am Batteriefach können die vier Batterien (AA) eingelegt werden. Leider verfügt das Messgerät nicht über eine externe Stromversorgungsbuchse. Durch längeres Drücken (>2 Sekunden) der roten Taste (An/Aus/Beleuchtung) wird das Gerät eingeschaltet. Im oberen Teil des Displays werden Datum/Uhrzeit angezeigt und im unteren Teil wird der Messwert in der Standardeinheit $\mu\text{Sv/h}$ sowohl als Zahlen als auch als Balkendiagramm dargestellt.

Die drei gelben Tasten rund um die rote Taste dienen zur Auswahl der Messmethode. Die gelbe, mit „SV/h – R/h“ beschriftete Taste links unterhalb des Einschaltknopfes schaltet zwischen der neueren Einheit Sievert/h und der älteren Einheit Rem/h um. Drückt man die oberhalb des Einschaltknopfes gelegene gelbe Taste „Sv“, arbeitet das Gerät als Dosimeter. Mit der rechts unterhalb des Einschaltknopfes befindlichen

gelben Taste „CPS – CPM“ aktiviert man den Modus zum Zählen der Impulse, wobei CPS die Anzahl der Impulse pro Sekunde und CPM die Anzahl der Impulse pro Minute zählt.

Kommen wir kurz zum Sensorkopf, in dem sich laut Internetrecherche ein hochwertiges Zählrohr der Firma LND Inc. aus New York (LND 712) befindet. Über einen drehbaren und eindeutig gekennzeichneten Sensorkopf ist beim DT-9501 die Art der Strahlungsmessung über eine Revolverblende einfach wählbar. Im Modus für Alpha-, Beta- und Gammastrahlung liegt das Zählrohr offen. Dreht man den Sensorkopf vorsichtig in die Stellung für Beta- und

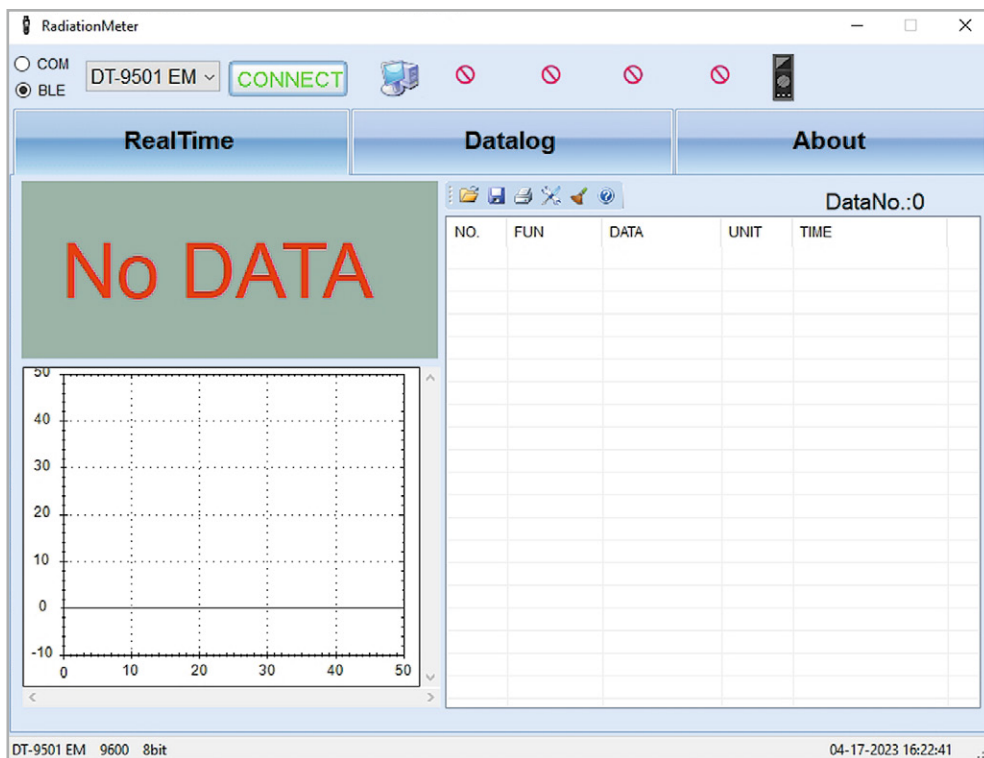


Bild 2: Startbildschirm der PC-Software

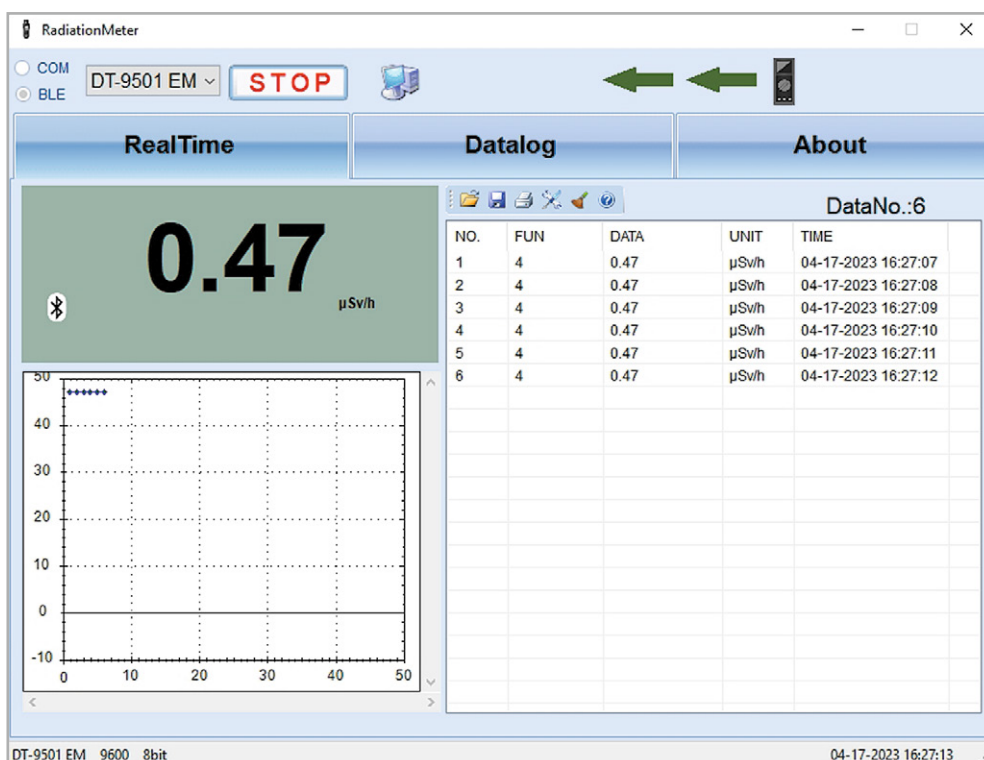


Bild 3: Erfolgreiche Datenübertragung und Anzeige

Gammastrahlung, wird eine dünne Blende oder Folie vor das Zählrohr geschoben, die Alphastrahlen abschirmt. In der dritten Stellung für Gammastrahlung wird eine dickere Blende vor das Zählrohr geschoben, die Alpha- und Betastrahlung abschirmt. Um Beschädigungen zu vermeiden, sollte man niemals etwas in die Öffnung des Zählrohrs einführen.

Im Menü kann neben dem Datum und der Uhrzeit ein Alarmgrenzwert eingestellt werden. Ebenfalls lässt sich ein Impulston (TICKERGERÄUSCH) aktivieren und die Bluetooth-Datenübertragung kann aktiviert oder deaktiviert werden.

Nach dem ersten positiven Eindruck von dem Messgerät waren die Erfahrungen mit der Software nicht so gut. Erstens liegt diese nur in Form einer CD vor, wobei moderne Laptops kaum mehr über CD-Laufwerke verfügen. Außerdem war die Software auf der CD in der Version 2.1 nicht mehr aktuell. Hier wäre es wesentlich sinnvoller, einen Link zu einer Downloadquelle für die aktuelle Version im Handbuch abzu drucken. Auf Nachfrage bei CEM Deutschland wurde mir gesagt, dass die aktuellste Software die Version 2.4 ist, und der folgende Link zur Verfügung gestellt [3]. Im Internet konnte ich herausfinden, dass es baugleiche Geräte mit der Bezeichnung PCE-RAM 10 und Reed R8008 gibt. Unter [4] gibt es für das R8008 ebenfalls die aktuelle Version zum Download. In der Bedienungsanleitung wird auf eine „software operation instruction“ verwiesen, eine solche Anleitung findet man leider weder auf der CD noch im Internet.

Deshalb möchte ich hier kurz beschreiben, wie eine Verbindung zwischen der PC-Software und dem DT-9501-Gerät zustande kommt: Dazu muss man wissen, dass es eine ältere Version mit klassischem Bluetooth gibt, die aktuelle Version jedoch mit Bluetooth Low Energy (BLE) arbeitet. Alle nun im ELVshop angebotenen Geräte verfügen über das BLE-Modul, deshalb beschränke ich mich auf die Beschreibung der BLE-Version. Von außen oder an der Modellnummer ist nicht ersichtlich, um welche Bluetooth-Variante es sich handelt.

Zuerst einmal muss Bluetooth im Menü des DT-9501 aktiviert werden (Drücken von Set-up, navigieren zum Bluetooth-Symbol, bis dieses blinkt, Enter drücken, On auswählen, mit Enter bestätigen und dann die ESC -Taste drücken). Im Display wird dann das Bluetooth-Symbol angezeigt. Danach startet man die Software „RadiationMeter“, die sich wie in Bild 2 zu sehen präsentiert.

Man wählt statt „COM“ links oben die Option „BLE“ und wählt dann im Dropdown-Bereich – wie in Bild 2 gezeigt – das DT-9501 EM aus. Wenn man auf die grüne Schaltfläche „Connect“ drückt, sollte die Darstellung wie in Bild 3 gezeigt zu sehen sein.

Mir ist noch ein kleiner Bug bei der Gerätesoftware aufgefallen: Nach einem Wiedereinschalten des Geräts zeigt dieses plötzlich das Bluetooth-Symbol nicht mehr an und man kann auch keine Bluetooth-Verbindung mit dem Gerät aufbauen, obwohl im Menü nach wie vor Bluetooth als „On“ markiert ist. Beheben lässt sich dies, wenn man noch einmal im Menü Bluetooth auswählt und mit der Enter-Taste die Auswahl „On“ bestätigt.

Da man mit der Software auf die PC-Umgebung unter Windows angewiesen ist und die Software nicht sehr komfortabel ist, habe ich versucht, mit einem ESP32-Board, das Bluetooth Low Energy beherrscht, ein Gerät zu bauen, das die Daten auf einer Webseite eines beliebigen Browser anzeigen kann. Dies war schwieriger als erwartet – Reverse Engineering war angesagt!

Webserver mit ESP32

Zuerst musste ich klären, welchen Bluetooth-Standard das Gerät verwendet und wie die Übertragung an sich funktioniert. Hilfreich erwies sich für die Lösung dieser Aufgabe eine App am Mobiltelefon namens „Bluetooth Scanner“ [5], die kostenfrei aus dem Google-Playstore zu beziehen ist. Die Darstellung des Messgeräts DT-9501 schaut dann wie in Bild 4 aus.

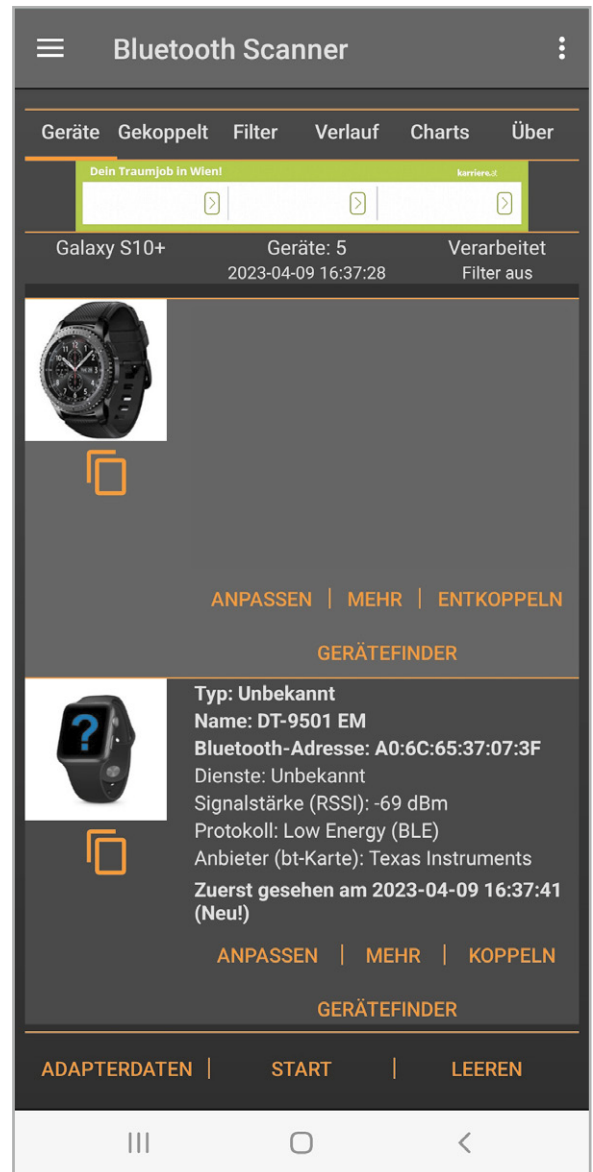


Bild 4: Anzeige des DT-9501 in der Bluetooth-Scanner-App

Dieser Bluetooth-Scanner zeigt bereits interessante Details: Er gibt den Bluetooth-Namen „DT-9501 EM“ an, die Bluetooth-Adresse und ganz wichtig das Bluetooth-Protokoll. Es handelt sich um das modernere und stromsparende Bluetooth-Low-Energy-Protokoll, der Bluetooth-Chip stammt von Texas Instruments. Diese Information ist wichtig, da klassisches Bluetooth und BLE zwei völlig getrennte Datenstacks und Protokolle verwenden und dies für die Implementierung im ESP32 von entscheidender Bedeutung ist.

Bevor ich den nächsten Schritt beschreibe, noch ein kurzer Ausflug in die Welt von Bluetooth Low Energy (BLE). BLE benötigt bis zu 100-mal weniger Strom als das klassische Bluetooth und wurde vor allem für Internet-of-Things-Geräte (IoT-Geräte) entwickelt. BLE unterscheidet zwischen einem Server und einem Client, wobei ein ESP32 beide Rollen einnehmen kann.

In unserem Fall ist das DT-9501-Messgerät der BLE-Server und der ESP32 der BLE-Client. Der BLE-Server wird für BLE-Clients in einem Prozess, der als „Advertising“ bezeichnet wird, sichtbar und sendet

Daten, die vom Client empfangen werden können. Der Client scannt naheliegende Geräte und falls er jenen BLE-Server findet, den er sucht, baut er eine

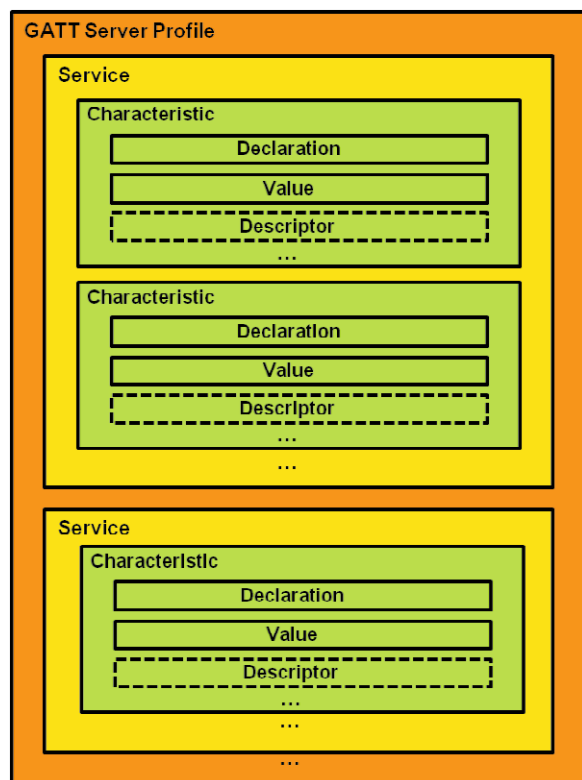


Bild 5: GATT-Modell [6]

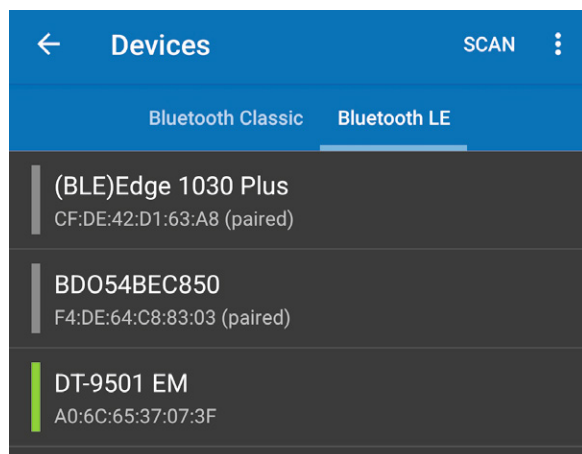


Bild 6: Auswahl des DT-9501 in der App Serial Bluetooth Terminal

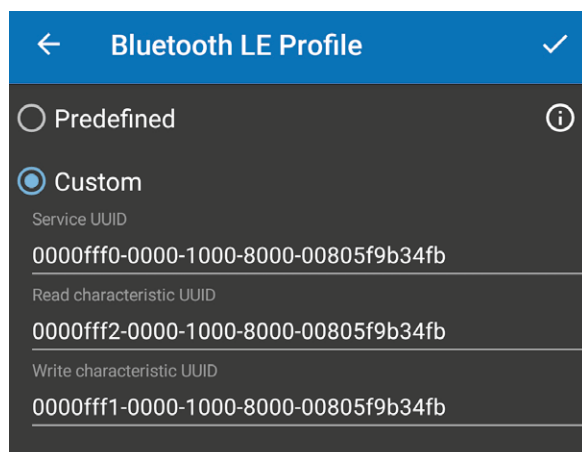


Bild 7: Custom-Einstellungen mit Anzeige der UUIDs

Verbindung auf und empfängt die gesendeten Daten. Die Datenstruktur entspricht dem sogenannten GATT-Modell (GATT steht für Generic Attribute) als eine hierarchische Datenstruktur wie in Bild 5 gezeigt.

Ein GATT-Server-Profil spezifiziert eine Standardsammlung von Diensten für einen spezifischen Anwendungsfall. Ein Service ist eine Sammlung von Informationen wie beispielsweise Sensordaten, Puls ... Eine Characteristic ist der Ort, wo sich die aktuellen Daten (Value) einer Hierarchie befinden. Ein Descriptor beschreibt optional Metadaten. Declaration beschreibt, mit welcher Methode der Wert (Value) abgefragt werden kann. Beispiele sind Write, Notify, Broadcast, Indicate ... Jeder Service und jede Characteristic besitzen eine UUID (Universally Unique Identifier). Eine UUID ist eine einzigartige 128-Bit-Zahl (16 Bytes), die von der Bluetooth Special Interest Group (SIG) vergeben wird.

Um mit dem ESP32 eine Verbindung zum DT-9501 aufbauen zu können, benötigt man die UUID für den Service, die UUID für die Characteristic sowie für die Declaration. Um diese Informationen herauszufinden, kommt eine weitere App mit dem Namen Serial Bluetooth Terminal [7] zum Einsatz, die ebenfalls im Google Playstore kostenfrei verfügbar ist. Unter „Devices“ wählt man das „DT-9501 EM“ aus, das sich im Reiter „Bluetooth LE“ befindet, wie in Bild 6 gezeigt.

Danach drückt man lange auf den Eintrag des DT-9501 EM, bis ein Auswahlmenü erscheint. Hier wählt man „Edit“ aus und klickt auf „Custom“, wie es Bild 7 zeigt.

Wir benötigen die „Service UUID“ für Service und die „Read characteristic UUID“ für die Characteristic. Diese muss später im Arduino Sketch eingetragen werden. Als Nächstes stellt man in der App unter Settings den „Display mode“ von „Text“ auf „HEX“. Dann erhält man die Darstellung wie in Bild 8, nachdem man unter Terminal den Button „Connect“ gedrückt hat.

Für die Declaration verwendet man Notify. Aus der Art, wie die Zeichen empfangen werden, kann man Folgendes schließen: Ein Datensatz beginnt mit D5 und endet mit 0D und besteht aus jeweils 15 Zeichen, die jede Sekunde unaufgefordert gesendet werden. Die Bedeutung der einzelnen Zeichen habe ich über weitere Versuche (ohne Gewähr) herausgefunden:

- Ein Datensatz beginnt immer mit einem Header von 4 Byte (HEX D5, F0, 00, 0A)
- Das fünfte Byte sind die beiden letzten Stellen des Jahres
- Das sechste Byte ist der Monat
- Das siebte Byte ist das Datum
- Das achte Byte ist die Stunde
- Das neunte Byte ist die Minute
- Im zehnten Byte sind jeweils 4 Bit für das vierte Digit und 4 Bit für das dritte Digit des Messwerts enthalten
- Im elften Byte sind jeweils 4 Bit für das zweite Digit und 4 Bit für das erste Digit des Messwerts enthalten
- Das zwölfte Byte ist immer 0
- Das dreizehnte Byte gibt die Position des Dezimalpunkts an
- Das vierzehnte Byte beschreibt den Messmodus
- Das fünfzehnte Byte ist immer HEX 0D

Mit diesen Informationen wurde ein Prototyp wie folgt aufgebaut. Zum Einsatz kam ein ESP32-Pico-Kit [8], das gut dokumentiert und leicht verfügbar ist und viele GPIOs bietet. Zur optischen Kontrolle habe ich lediglich eine rote LED an GPIO21 und eine grüne LED an GPIO22 über jeweils einen 150-Ω-Widerstand an Masse gelegt. Die rote LED zeigt an, dass eine Bluetooth-Verbindung zum DT-9501 fehlt, und die grüne LED zeigt eine erfolgreiche Verbindung zum DT-9501 an.

Damit komme ich zur kurzen Beschreibung der Demosoftware. Diese wurde in der Arduino IDE 1.8.16 erstellt und kann unter [9] heruntergeladen werden. Wichtig ist, dass die Board-Erweiterung für den ESP32 installiert ist; ich habe die Version 1.0.6 verwendet. Diese beinhaltet schon die BLE Libraries. Als einzige externe Libraries müssen

„ESPAsyncWebServer“ und „AsynTCP“ installiert werden, die man unter [10] und [11] herunterladen kann und die in das „libraries“-Verzeichnis der Arduino-Installation gehören.

Als Board muss das „ESP32-Pico-Kit“ und - ganz wichtig - unter „Partition Scheme“ der Eintrag „No OTA (Large APP)“ ausgewählt werden. Danach sollte die Software kompilierbar sein. Vor der Übertragung auf das Board sollten noch das „BLE-Remote-Service-UUID“ in Zeile 61 und die „Characteristic-UUID“ in Zeile 63 dahingehend überprüft werden, ob sie mit den Werten aus der App übereinstimmen.

Obwohl der Sketch ausführlich kommentiert ist, möchte ich kurz auf die Funktionsweise der Software und deren Bestandteile eingehen. Die Software spannt zuerst einmal einen eigenen WiFi-Access-Point mit der SSID „DT-9501“ und der IP-Adresse „192.168.1.4“ auf. Danach werden die auffindbaren BLE-Server aufgelistet und nach dem DT-9501-Messgerät gesucht. Ist dieses gefunden, verbindet sich der ESP32 mit diesem unter Verwendung der Service- und Characteristic-UUID und eine Callback-Routine für Notify wird aufgesetzt. War dies erfolgreich, wird auf den Empfang von Messwerten gewartet und ein empfangener Datensatz von 15 Byte wird ausgewertet. Drei Strings, die das Datum und die Zeit des Empfangs enthalten (Datum und Zeit stammen dabei vom Messgerät), sowie der String, der den Messwert samt Einheit enthält, werden an eine Webseite übergeben, die aufgerufen wird, sobald die IP-Adresse „192.168.1.4“ im Browser eingegeben wird. Diese Webseite wird vom ESP32 asynchron zum restlichen Code abgearbeitet. Nach dem Übertragen des Sketches auf das ESP32-Pico-Kit sollten Nachrichten im Serial Monitor der Arduino IDE wie in Bild 9 gezeigt erscheinen.

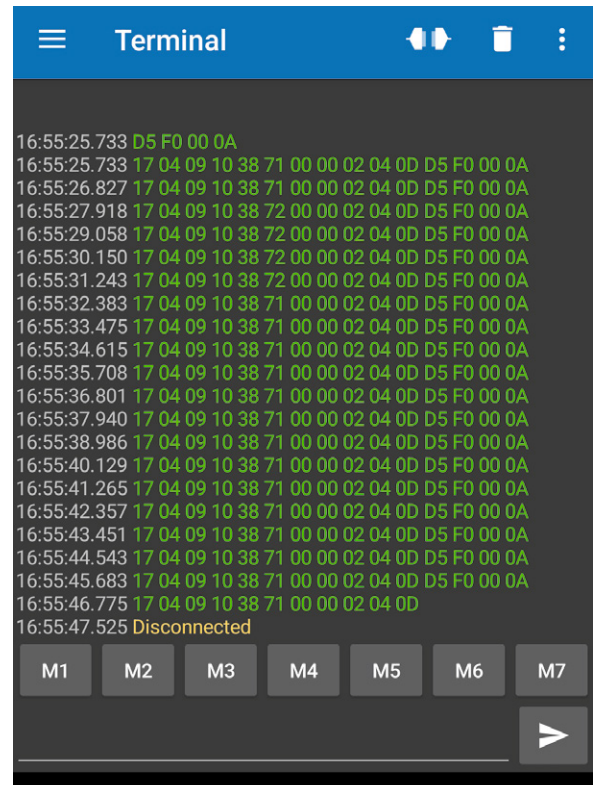


Bild 8: Empfang von Zeichen vom DT-9501 in der App Serial Bluetooth Terminal in HEX

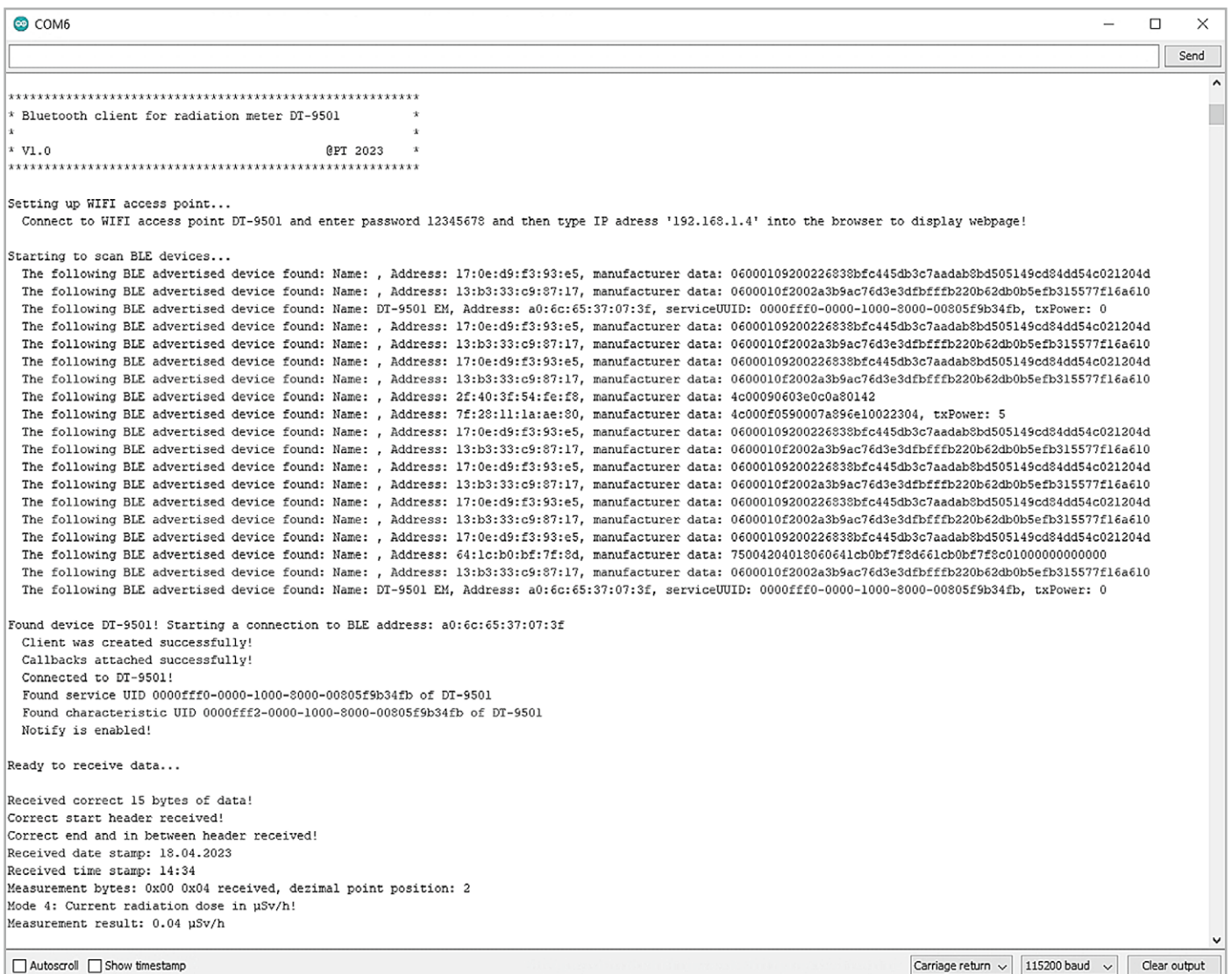


Bild 9: Ansicht des Startvorgangs im Serial Monitor der Arduino IDE

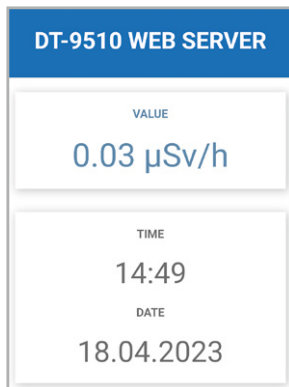


Bild 10: Webseite mit Anzeige der Messdaten

Nach dem Begrüßungstext wird der WiFi-Access-Point aufgebaut und dann werden die BLE-Server in Reichweite gescannt. Sobald das DT-9501 gefunden wurde, werden die Initialisierungen bzw. die Verbindung zu diesem aufgebaut. Danach wird ein erster Datensatz empfangen und dekodiert. Verbindet man sich nun beispielsweise mit einem Mobiltelefon mit dem WiFi-Access-Point des ESP32 und gibt danach die Adresse 192.168.1.4 ein, wird eine Webseite wie in Bild 10 zu sehen angezeigt.

Noch ein paar Anmerkungen zum Arduino-Sketch: Im Header zum Sketch befinden sich zwei Links zu Webseiten, die tiefer auf das Thema BLE eingehen, das in der Arduino IDE recht unübersichtlich integriert ist.



Bild 11: Quader aus Uranglas

In Zeile 50 können durch Auskommentierung die Debug-Meldungen ausgeschaltet werden. Ab Zeile 87 können die Parameter für den WiFi-Access-Point angepasst werden. Ab Zeile 106 ist der Code für die Webseite eingefügt, der ebenfalls angepasst werden kann. Danach folgen die Bluetooth-Callback-Funktionen, wobei ab Zeile 216 die Notify-Routine implementiert ist. Ab Zeile 255 findet man jene Funktion, die die aktuellen Messwerte in die Webseite einbaut. Danach folgt die Funktion, um sich per BLE mit dem DT-9501 zu verbinden.

Die Funktion „SUB_DECODE_MESSAGE“ ab Zeile 355 dekodiert das empfangene Datenpaket und übergibt die drei Strings an den Webprozessor. Ab Zeile 624 folgt die Set-up-Routine, in der zuerst die Pins und die serielle Schnittstelle konfiguriert werden. Danach folgen der WiFi-Access-Point sowie das Aufsetzen der BLE-Routinen. Die Hauptroutine ist recht kurz. Sie wartet zuerst, bis das DT-9501 verbunden ist, und danach auf empfangene Daten, die zu dekodieren sind.

Test des DT-9501

Ein Test des DT-9501 ist gar nicht so einfach, da man ja dafür weder nach Tschernobyl reisen noch auf einen Atomunfall warten möchte. Eine einfache Möglichkeit bietet Uranglas [12], das eine Uranoxidverbindung enthält, die das Glas hellgelb bis hellgrün anfärbt und unter UV-Licht schön schimmert. Dieses wurde in größeren Mengen um 1940 in Böhmen hergestellt und war angeblich auch schon den Römern bekannt. Auf ebay erstand ich einen 8 x 8 x 2 cm großen Quader (Bild 11), der extra zum Testen von Geigerzählern angepriesen wurde.

Der größte Teil der Strahlung, die von Uranglas ausgeht, ist Betastrahlung mit einem kleinen Alphaanteil. Die Strahlung sollte allerdings nur direkt an der Oberfläche messbar sein.

Im Bereich für Alpha-, Beta- und Gammastrahlung habe ich direkt an der Oberfläche 1,5-1,7 µSv/h gemessen. Noch deutlicher ist der Unterschied zur normalen Umgebung im Modus der Impulszählung: Hier sieht man recht schön, um wie viel schneller das Messgerät hochzählt, je näher man es an die Glasoberfläche hält. Im Bereich für Beta- und Gammastrahlung geht die Anzeige auf ca. 1-1,2 µSv/h zurück, ein Beweis für einen Anteil an Alphastrahlen. Im Bereich Gammastrahlung gibt es kaum eine Erhöhung bezüglich des Normalwertes.

Ich hoffe, dass ich Ihnen mit diesem Beitrag einen Einblick in die spannende Welt der Radioaktivitätsmessung mit dem interessanten und hochwertigen Gerät DT-9501 von ELV geben und Ihnen zeigen konnte, wie man sich an das „Reverse Engineering“ herantastet, um Geräte in eigene Applikationen einzubinden.

Die ESP32-Demosoftware stellt eine Basis für weitere individuelle Anwendungen und Experimente dar. **ELV**

i Weitere Infos

- [1] https://www.bfs.de/DE/home/home_node.html
- [2] CEM Radioaktivitätsmessgerät DT-9501: Artikel-Nr. 252804
- [3] https://www.dropbox.com/s/yk7u91zhwd3eryz/RadiationMeter_V2.4_%E6%97%A0%E5%94%9B_20220616.zip?dl=0
- [4] <https://www.reedinstruments.com/content/software-drivers>
- [5] <https://play.google.com/store/apps/details?id=com.pzolee.bluetoothscanner&hl=de>
- [6] Quelle: <https://microchipdeveloper.com/wireless:ble-gatt-data-organization>
- [7] https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=de
- [8] ESP32 Pico Kit:
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-pico-kit.html>
- [9] Downloadverzeichnis: Artikel-Nr. 253604
- [10] ESPAsyncWebServer: <https://github.com/me-no-dev/ESPAsyncWebServer>
- [11] AsyncTCP: <https://github.com/me-no-dev/AsyncTCP>
- [12] <https://de.wikipedia.org/wiki/Uranglas>

Alle Infos finden Sie auch online unter: de.elv.com/elvjournal-links