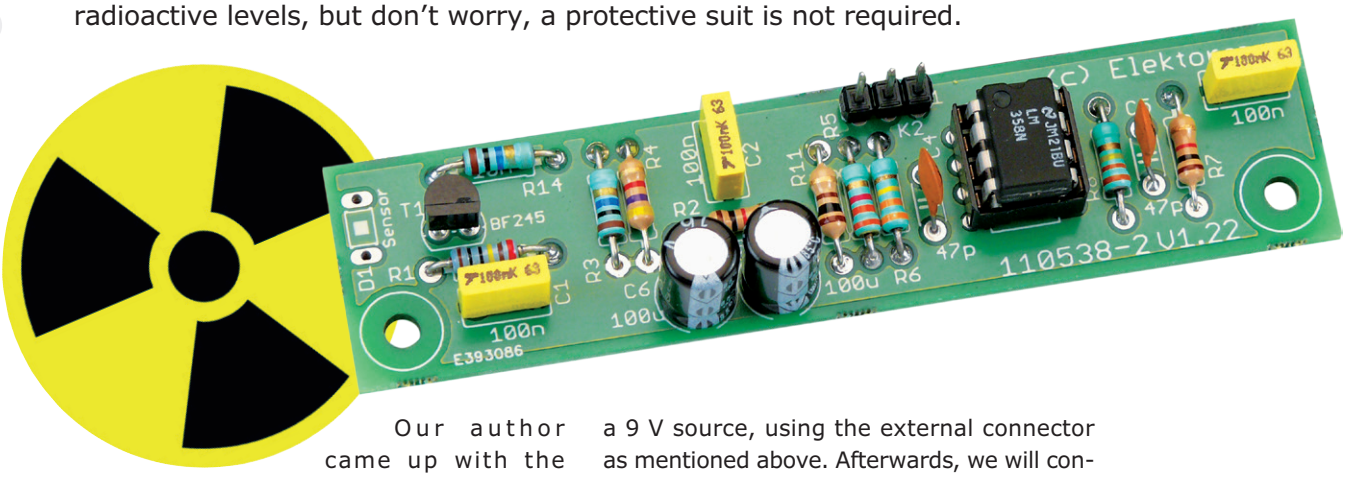
An Arduino'd Radiation Meter

Now that's a *radiant* idea

By
Alfredo Bertero
(Italy)

These days no one marvels at seeing an Arduino board in just about any application. Arduino's ease of use makes it perfect to quickly develop DIY measuring instruments. This article shows how to build a simple meter for radioactive levels, but don't worry, a protective suit is not required.



Our author came up with the idea of connecting a radiation meter probe to an Arduino UNO when he realized that this board has two options in terms of power source. We have the standard 5 V coming through the USB connection, while alternatively it's possible to use an external power source up to 12 V via the 2.1 mm jack connector. When connecting both sources at the same time (USB and external), Arduino automatically selects the external one. Besides, the selected power source will be available at V_{in} (see "power" pins on your board).

Tackle the h/w first...

Let's proceed with the hardware configuration. Apart from the external power source, for our setup we also need a radiation probe, such as the brilliant one used in the Elektor project "Improved Radiation Meter" [1]. The meter board design may be downloaded free of charge, as well as ordered ready-made, at this website.

First of all, we have to power the board from a 9 V source, using the external connector as mentioned above. Afterwards, we will connect the radiation sensor module power wire K2.3 to the V_{in} pin (9 V) on our Arduino UNO board, and K2.1 to GND. The signal wire of the probe (K2.2) should be connected to the Arduino's A0 analog input pin, as depicted in **Figure 1**. A piezo buzzer was also connected to the digital (PWM) pin 12 and GND. **Figure 2** shows our setup ready to act.

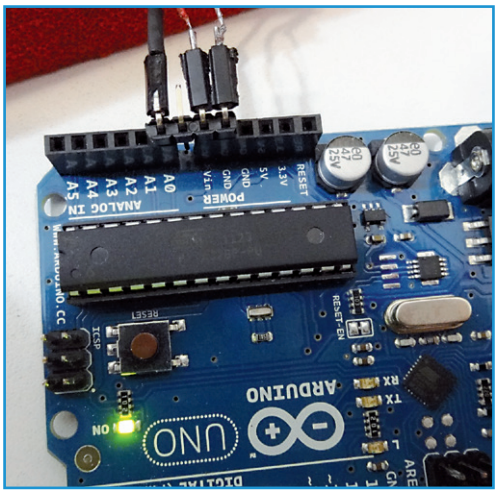


Figure 1.
Connecting the probe to your UNO board.

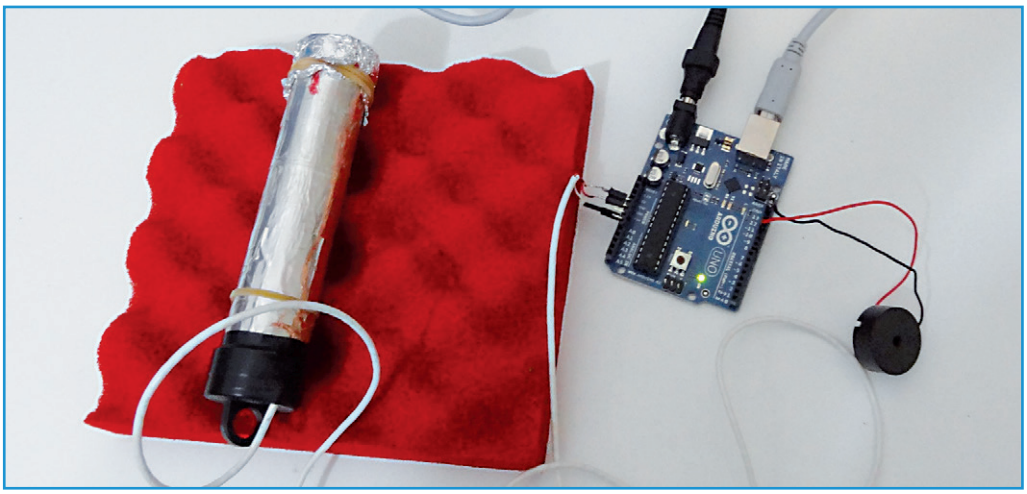


Figure 2.
The required setup is quite simple.

And then get down to the code

As with every Arduino project, your code should be in form of a sketch (this is how Arduino programs are called and have an ".ino" file extension). This firmware consists of a translation of the original code of the Elektor's Improved Radiation Meter, using the Arduino environment (IDE), which is available free of charge at the Arduino website [2]. As a result, two versions of the firmware are provided, which can be read with the notepad or the Arduino environment itself. You may download them at the web page for this article [3]. Besides, additional information may be found posted at Elektor**.**LABS [4].

The first version (radiationElekt.ino) transfers each value read from the probe through the Arduino serial port, and also calculates the CPM (Counts Per Minute) rate. To check the values read, and to regulate the threshold, the program has been tested with the Arduino serial monitor (in the Arduino IDE, it can be found under Tools -> Serial Port). However, a simple terminal program may also be used for this purpose.
In this version, at first the average noise level is calculated by means of 2,000 samples. Later, together with the user defined variable 'Threshold', the minimum peak value is set. In case you want to modify this threshold, it has to be done manually in the Arduino sketch, and loaded again into our UNO board, which takes just few seconds.

The number of counts is output every 10 seconds. Then, this number of peaks is multiplied by 6, providing the total value of counts per 60 seconds, i.e. the CPM (Counts Per Minute). Apart from providing the peaks via the serial output, the LED will flash briefly with each one, and a 5-Hz crackle is generated using the buzzer.

The second version (radiationElekt2_VB.ino) provides an output using the Arduino virtual serial port with values read from the probe through the A0 analog input pin. This sketch can be used with the VB Spectrum program, plotting the histogram of the instant values of the radiation. The Spectrum Meter software is available at the website noted above [1].
Figure 3 shows a screenshot of the VB Spectrum Meter software representing the normal background radiation noise within 15 minutes.
Figure 4 represents what happens when putting the probe close to a cuprosklodowskite [5] sample, a uranium-ish mineral notable for its gamma radiation "L" corresponds to the user defined threshold, while "t" is the time in seconds.

At that point, the values are only provided via the serial port, but adding a display to your Arduino board should not be difficult at all, as there's a ready-made library available at the Arduino community [6].
It has to be noted that this setup also works if it's powered from the USB port only, even though it's not recommended. The meter will be less sensitive and you may need to lower the threshold value.

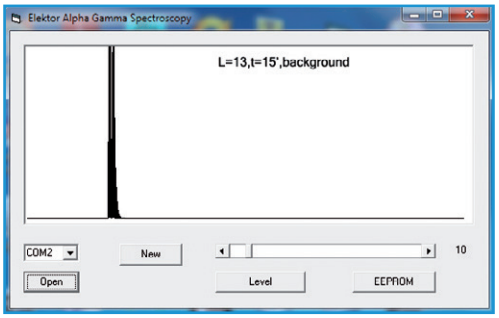


Figure 3.
VB Spectrum software showing the normal background radiation noise.

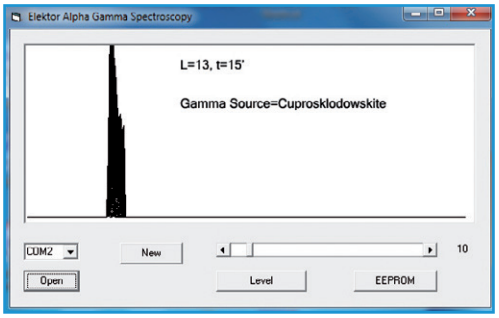


Figure 4.
Results from using a sample of uranium mineral.

Finally, it is important to mention that the baud rate set on the Arduino board matches that of the serial terminal program.
Now catching those particles and rays should be child's play (seriously, it shouldn't).

(120468)

Internet Links

[1] www.elektor.com/110538

[2] http://arduino.cc/en/main/software

[3] www.elektor.com/120468

[4] www.elektor-labs.com/ElektorPOST/2013/06

[5] http://en.wikipedia.org/wiki/Cuprosklodowskite

[6] http://arduino.cc/en/Reference/LiquidCrystal