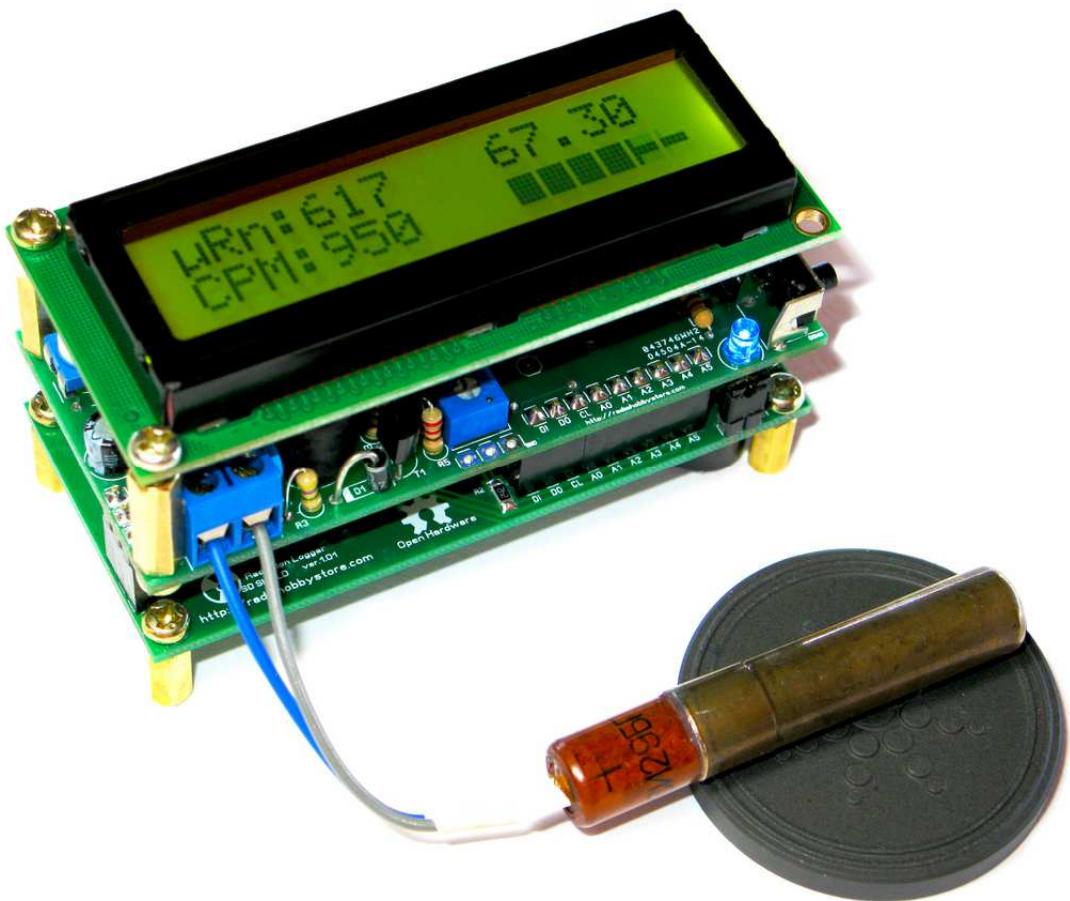


# Arduino IDE Geiger SD Logger

## DIY Kit ver. 1.01

RH-K-GK-1-L



<http://rhelectronics.net>

This is open source code Geiger project based on Arduino IDE manufactured by RH Electronics <http://rhelectronics.net> The project has two models: main board only and main board with SD logger shield. **The document you are reading now describes main board model with SKU RH-K-GK-1-L.**

**RH-K-GK-1-A:** kit includes main board with LCD. This model compatible with 400V-500V GM tubes. Cheap solution for computer monitoring nuclear radiation station or for portable Geiger counter. Buzzer is not included on main board pcb, but connection of external buzzer is supported to produce clicker sound similar to a "classic" Geiger counter sound. Support eeprom logging of absorbed dose.

**RH-K-GK-1-L:** kit includes main board with LCD and Real Time Clock with SD logger shield. This model compatible with 400V-500V GM GM tubes. Kit includes SD logger board and has many advantages for logging radiation data on micro SD with or without GPS. Buzzer is included on shield board and produce clicker sound similar to a "classic" Geiger counter sound.



**Please consider the SKU model when you purchase!**

Take note, the project IS NOT PIN TO PIN COMPATIBLE with BroHogan!

Unlike Brohogan's kit, we excluded 555 timer and 74hc14 to reduce components, because it possible to produce high voltage with PWM and led flashing with just another IO. The project has main PCB and SD board shield and it possible to use the main board only if needed. The primary idea was to create simple portable SD-card radiation logger based on open Arduino IDE. You are free to modify and improve the code under GNU license. Take note, you are responsible for any software and hardware modifications. Please send your suggestions or bug reports at support@rhelectronics.com

**High Voltage with PWM:** By default the main board designed to work with 400V-500V GM Tubes. You need to tune high voltage output through software. Microcontroller **not stabilizes** strictly the high voltage output, but it will correct high voltage if battery is half or low.

**Micro SD cart shield:** is the main benefit of this project. Up to 100 txt logs and 100 csv log files can be created on the card before you'll need to transfer the data to the computer. Additional file keep total absorbed dose you received during lifetime of this device. You can re-zero dose file manually or delete it for new count. The files on SD have following format:

LOG00.TXT

"MM/DD/YYYY,HH:MM:SS, minuteCpm CPM, absorbedDose uSV, CurrentBat mV"  
"\$GPRMC,041003.000,A,3147.1361,N,03512.8326,E,2.59,284.41,231213,,A\*63"

Only **Adafruit GPS** module currently can be connected to RX/TX pins. NMEA data is not parsed and written to LOG.TXT file if you enabled GPS support in Arduino sketch. Data written once per minute to the file. I would say it great for walking with Logger, but if you need faster nmea data saving you should modify the sketch to safe nmea in other periods.

LOG00.CSV

"MM/DD/YYYY,HH:MM:SS,CPM"

CSV file is compatible with RH Electronics "Radiation Logger" <http://www.rhelectronics.net/store/radiation-logger.html> You can open and view the graph in "Radiation Logger". Data written to the file every 10 seconds.

DOSE.TXT

This file contains total absorbed dose value in uSv units. Dose file updated once per minute. If you need to start new lifetime count you can write "0.00" to the file or delete it, the software will create a new file.

Recommended size for micro SD card is 2Gb - 8Gb.

**Real Time Clock** IC keeping the date and clock. RTC requires CR1220 battery (not included). PCB allows CR2032 / CR2025 battery support with different low cost battery holder. To set up your local time you'll need to upload RTC setup sketch first! See more technical details later in the manual. Take note, DS1307 is excellent RTC, but it loose minute or two during several days, so don't expect the clock to be super precision.

**Buttons functions** are simple. "Down" marked button switches between uSv/h and uRn/h units on the display. "Up" marked button will light up LCD backlight for small period and will show the clock instead of total absorbed dose.

**CPM calculation algorithm** is moving average with recalculation option for rapid changes. It still recommended to wait one minute for full measurement.

**UART logging** to computer via TTL-USB dongle supported if no GPS is connected. You can use our freeware "Radiation Logger" for Windows. For UART logging you can use the kit without SD shield, please discover Arduino sketch how to turn on or turn off "Radiation Logger" out.

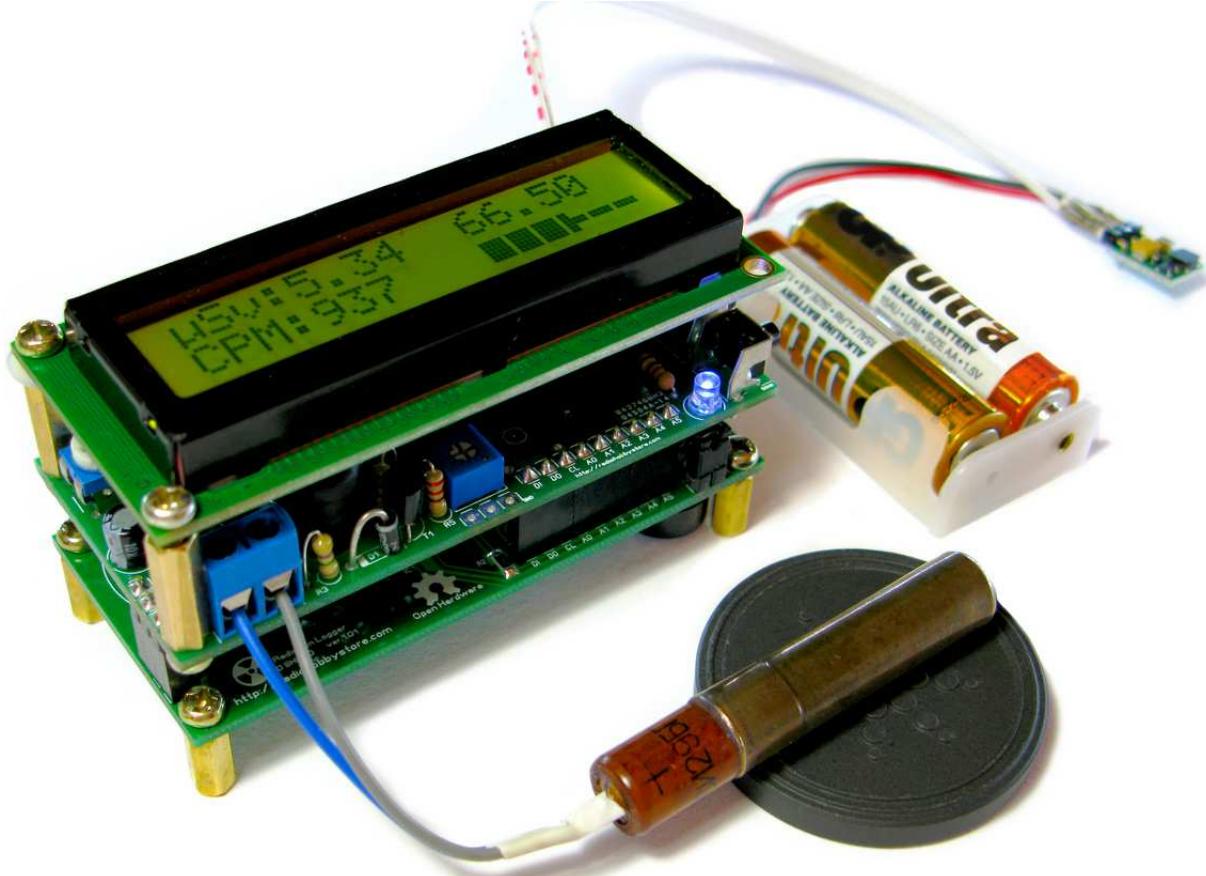
**Smart backlight control** helps to save battery. Microcontroller will light on the LCD backlight by auto if CPM value reach ALARM settings. LCD backlight driven through transistor key.

**Alarm pin** combined with smart backlight. BL pin can be connected to external alert system, but remember about Atmega port current limitation.

**Clicker sound** that similar to classic Geiger counters produced by buzzer. Additional 3-times short beeps produced when CPM level is high.

**Fast Bargraph** on LCD with scale, configurable through software. Bargraph shows CPS (count per second) measurement. Not useful for background radiation, but can be excellent for search mode.

**Battery indicator** for Ni-MH batteries will inform you if capacity is on half or low.



## Assembling of the Main Board:

Before you start soldering the main board please follow components list table. Main-Board parts are DIP through-hole easy to solder. Some components has polarity, please refer electrical circuit, pcb silkscreen and physical photos from this manual!

Please follow our assembling and soldering instruction steps. Package includes printed electrical circuit. Keep it in front of you during the soldering for easy reference.

Remember, do clean solder work and install right components in the right place. It is always hard to unsolder components from the PCB. We cannot be responsible if you'll overheat your PCB or will damage the kit components by incorrect soldering. Double check yourself before you solder. Use a solder with Rosin Core Flux.

Take your time when performing solder work. It may take about 2 hours to complete the kit, depend on your skills.

Do not apply too much solder. A good joint should look like:



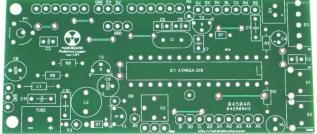
We advise you to use 0.6-0.8mm thickness lead solder wire with low melt point for this kit, such as:

60/40 – 186 Celsius (386 Fahrenheit)

63/37 – 183 Celsius (361 Fahrenheit)

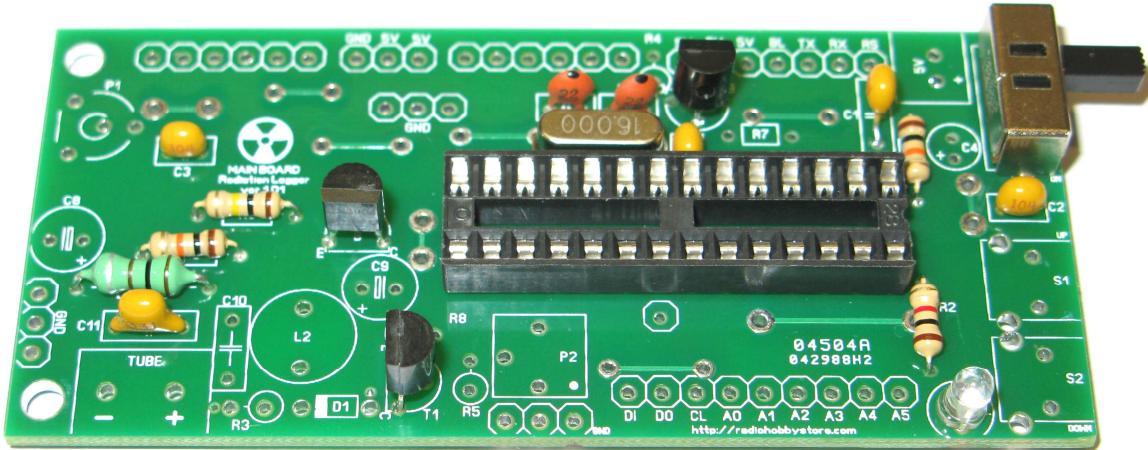
Image	Name	Quantity	Value	Description
	R8	1	100 ohm	¼ W resistor
	R4	1	220 ohm	¼ W resistor
	R2,R5	2	1K ohm	¼ W resistor
	R1, R6, R7	3	10K ohm	¼ W resistor
	R9	1	100K ohm	¼ W resistor
	R3	1	4.7M ohm	¼ W resistor
	P1	1	10K ohm	LCD contrast trimmer
	P2	1	1K ohm	HV Trimmer

	C1, C2, C3, C7	4	100nF	Ceramic Capacitor (104)
	C11	1	330pF	Ceramic Capacitor (331)
	C5, C6	2	22pF	Ceramic Capacitor (220)
	C10	1	10nF	HV Capacitor
	C4, C8, C9	3	100uF	Electrolytic Capacitor
	D1	1	1N4937	Fast Rectifier Diode
	LED1	1	LED	3mm red led
	T1	1	MPSA44 KSP44	HV NPN Transistor
	T2	2	2N3904	NPN Transistor
	L1	1	100uH	Axial Inductor
	L2	1	10mH	Radial Inductor
	Cr	1	16MHz	Crystal HS-49S
	IC1	1	Atmega 328A	Microcontroller with Bootloader

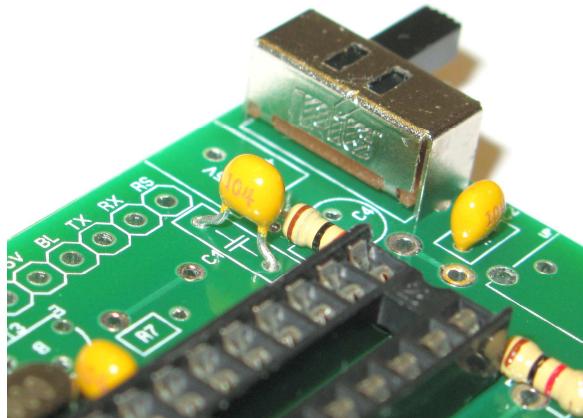
	<b>Socket</b>	<b>1</b>	<b>DIP 28</b>	<b>IC socket</b>
	<b>S3</b>	<b>1</b>	<b>Slide Switch</b>	<b>On/Off switch</b>
	<b>Connector</b>	<b>1</b>	<b>Terminal Block</b>	<b>Tube connector</b>
	<b>S1, S2</b>	<b>2</b>	<b>Tact Switch</b>	<b>Up/Down Buttons</b>
	<b>Connector</b>	<b>2</b>	<b>6 Pin Female</b>	<b>LCD connector</b>
	<b>Connector</b>	<b>6 Pins</b>	<b>Male Right Angle</b>	<b>Peripheral connector</b>
	<b>Connector</b>	<b>20 Pins</b>	<b>Male</b>	<b>Peripheral connector</b>
	<b>Connector</b>	<b>1</b>	<b>Male</b>	<b>5V Connector, check polarity and direction!</b>
	<b>Connector</b>	<b>12 Pins</b>	<b>Male Long</b>	<b>LCD Connector</b>
	<b>Standoff</b>	<b>2</b>	<b>M3 15mm</b>	<b>LCD Standoff</b>
	<b>Screw</b>	<b>4</b>	<b>M3 x6mm</b>	<b>LCD Screw</b>
	<b>LCD</b>	<b>1</b>	<b>16x2</b>	<b>HD44780 Green</b>
	<b>PCB</b>	<b>1</b>	<b>81x37 mm</b>	<b>Main Board</b>

Recommended to start with all horizontal placed resistors and L1 axial coil. Please use supplied printed electrical circuit you received with the package, and refer pcb silkscreen for components placement.

Put special attention for transistors placement and LED polarity. Short lead of the LED is a cathode (minus) side. Do not rely on resistors color code, recheck it with a multimeter.



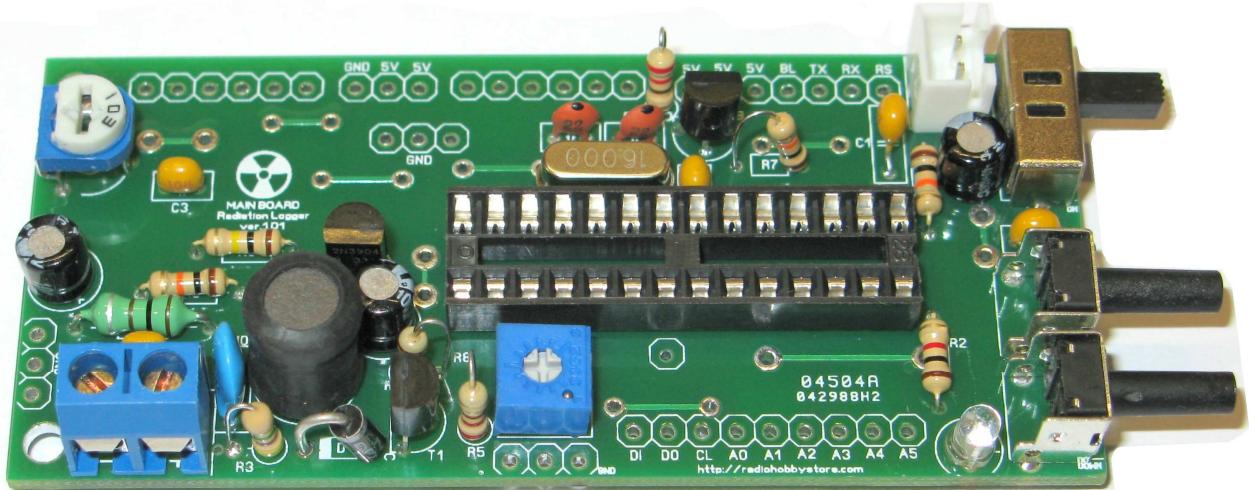
C1 capacitor has wide 5.0mm lead position; you can bend it with pliers. Other ceramic caps installed close to the pcb. Try to make all electrical contacts short as it possible.



D1 diode should be soldered with white strip toward R3.

R3 is GM tube load resistor and the value depends on Geiger tube type you'll use. For SBM-20 default 4.7M is suggested. Some tubes used with 2M or with 10M load resistor. Please verify with tube datasheet before soldering R3.

If you are going to use 400V-500V GM Tube then install D1 and R3 as pictures shows. For 600V mode skip now R3 installation and follow for 600V mode guide (discussed later in the manual).



**5V JST white connector has polarity! Positive side is on C4 capacitor side!**

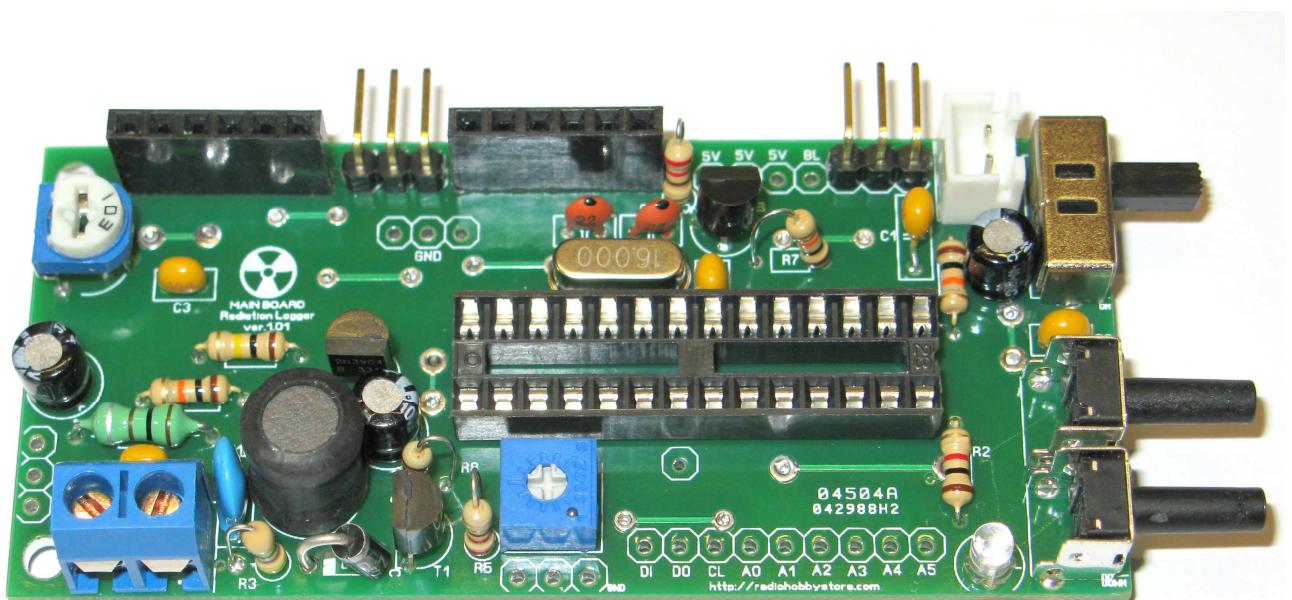
Some samples of JST connectors may have different orientation, be sure to check with PCB silkscreen (+) and connector red wire direction and polarity!

P1 is LCD contrast potentiometer. Trim it in anti-clockwise direction for adjusting display contrast.

P2 used for high voltage fine adjustment. It's recommended to trim it to the middle point and use software calibration for high voltage.

Electrolytic capacitors should be installed in proper polarity. All other capacitors are ceramic and don't have polarity.

2x6 pin female connectors for LCD are soldered on the pcb and 2x6 pin male pins soldered on LCD module.



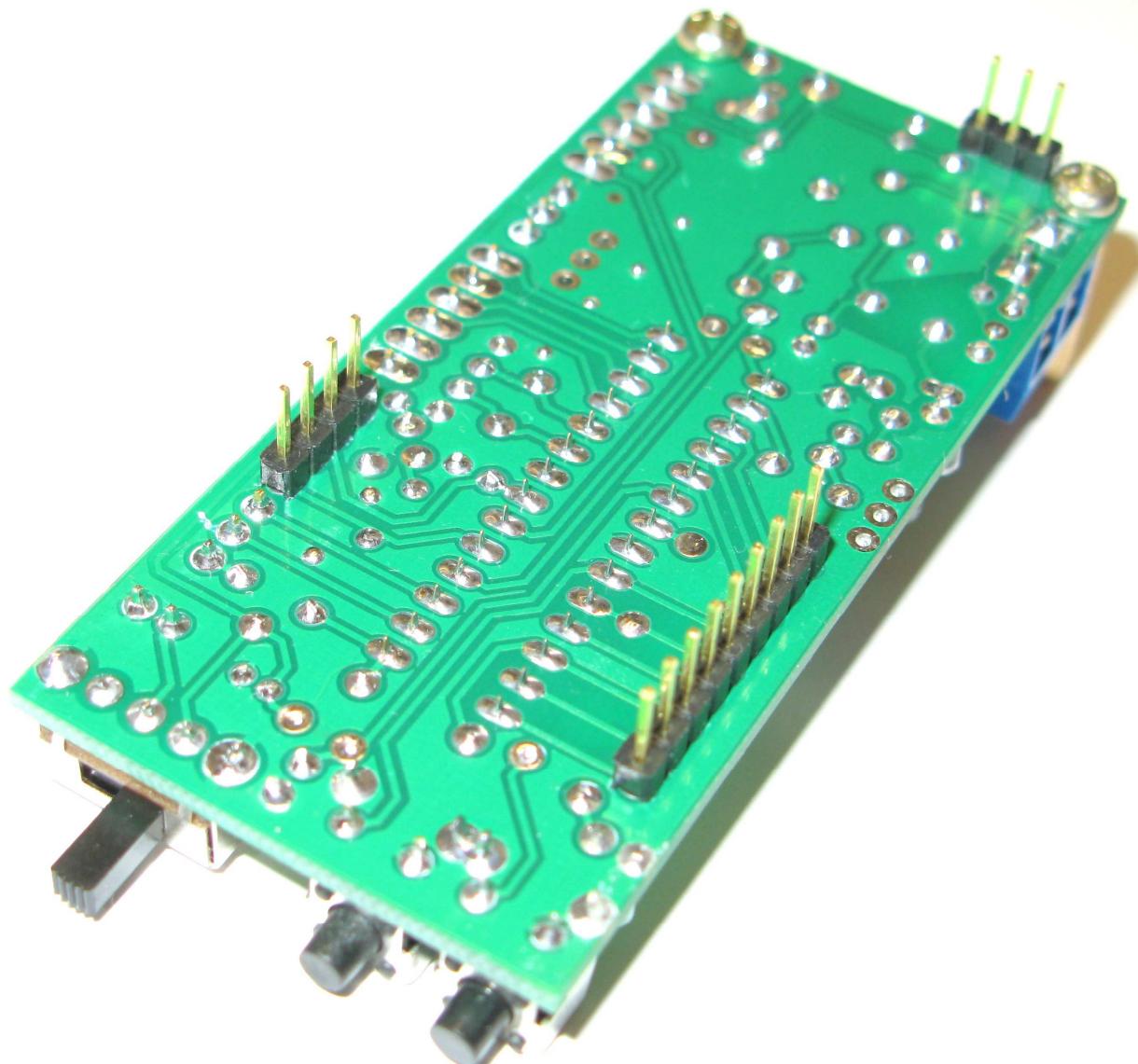
**Take note, ATMEGA pitch is toward buttons side! Wrong polarity may damage the chip!**

SD Shield pins connectors **for RH-K-GK-1-L** soldered on the bottom side of the pcb toward bottom position. These connectors soldered in case you are planning to connect SD shield! If you are going to use main board only then you can skip soldering these pins for now.

After soldering you need to wash the Main Board with Flux Remover (isopropyl alcohol) and soft brush. Do not allow isopropyl to split into mechanical parts such as buttons. It may create grease glue inside the buttons and prevent it to be pressed smoothly. If you have this issue then wash the board and buttons again until all flux will be deleted. Dry the board at least 30 minutes or more.

It's very important to clean the pcb properly. Any small flux remains will affect high voltage converter stage.

Inspect the board with magnifier to be sure you haven't shorts and flux. Test with multimeter tester that all 5V and all GND points are ok. Here is the example of the clean board with good solder joints. Verify your solder work quality.

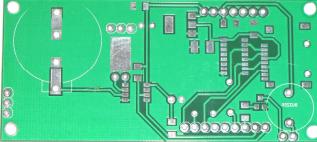


## Assembling of the SD Shield Board:

Before you start soldering the SD Shield board please follow components list table. SD-Board parts are SMT and require special skills for soldering. You can use hot air or solder with tiny tip. If you never soldered SMD parts before, we suggest you to read several guides, such as Sparkfun tutorial: <https://www.sparkfun.com/tutorials/96>

Some components has polarity, please refer electrical circuit, pcb silkscreen and physical photos from this manual!

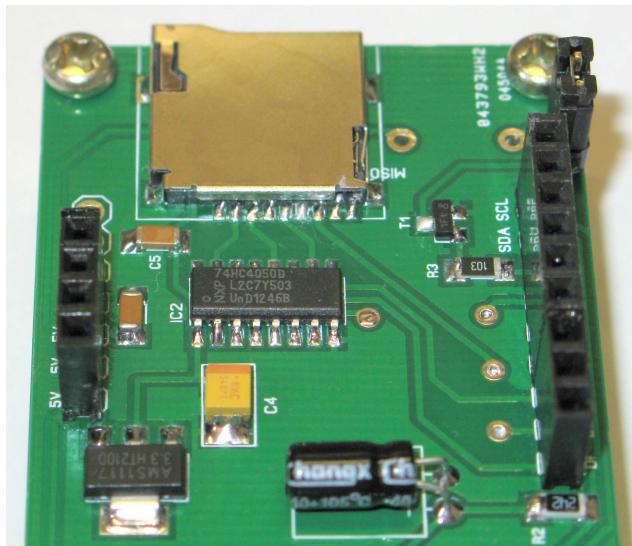
Image	Name	Quantity	Value	Description
	R1, R2	2	2.2K ohm	1206 SMT resistor (222)
	R3	1	10K ohm	1206 SMT resistor (103)
	C2, C3, C5	3	100nF	1206 SMT capacitor
	C4	1	10uF	3528 SMT Tantalum
	C1	1	100uF	Electrolytic Capacitor
	T1	1	BC847	SMT Transistor sot-23
	IC3	1	ASM1117-3.3V	3.3V regulator sot-223
	IC1	1	DS1307	RTC soic-8
	IC2	1	74HC4050	Level Shifter soic-16
	Cr	1	32.768KHz	Crystal
	Socket	1	Micro SD socket	SMT
	Socket	1	CR1220 Holder	Battery Holder SMT

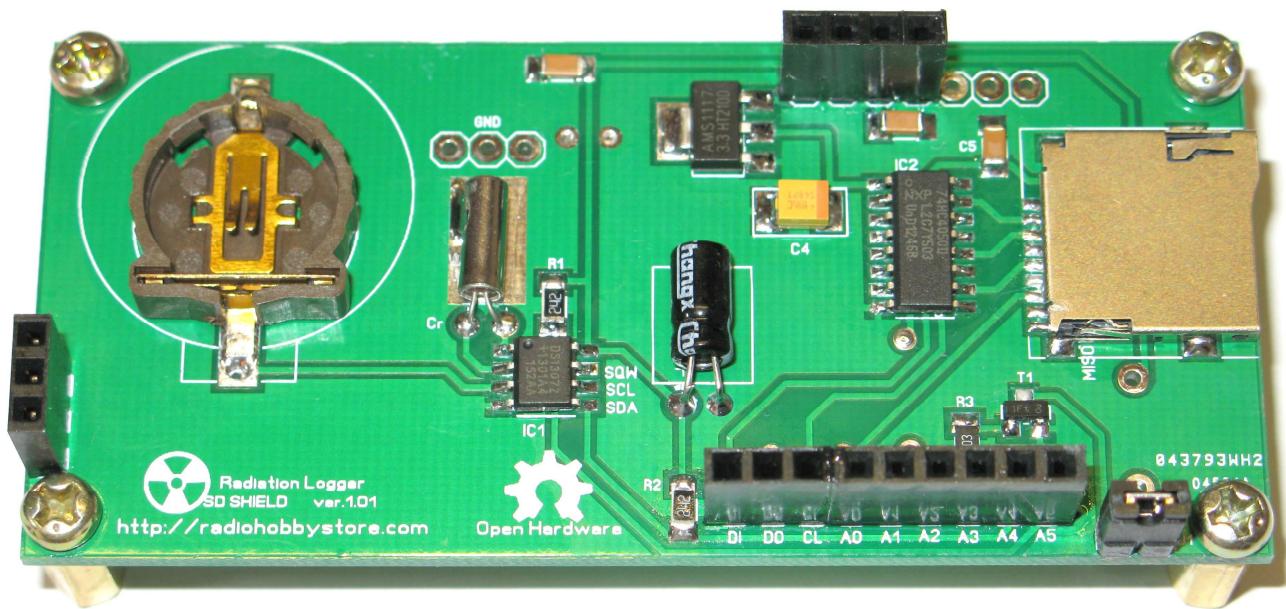
	<b>Connector</b>	<b>2</b>	<b>3 Pin Female</b>	<b>Shield connector</b>
	<b>Connector</b>	<b>1</b>	<b>4 Pin Female</b>	<b>Shield connector</b>
	<b>Connector</b>	<b>1</b>	<b>6 Pin Female</b>	<b>Shield connector</b>
	<b>Buzzer</b>	<b>1</b>	<b>5V</b>	<b>Piezo Buzzer</b>
	<b>JMP1</b>	<b>1</b>	<b>Jumper</b>	<b>Buzzer Jumper</b>
	<b>Standoff</b>	<b>4</b>	<b>M3 11mm</b>	<b>Shield Standoff</b>
	<b>Screw</b>	<b>4</b>	<b>M3 x6mm</b>	<b>Shield Screw</b>
	<b>PCB</b>	<b>1</b>	<b>81x37mm</b>	<b>Shield Board</b>

Recommended to start soldering 1206 resistors and capacitors first. Additional attention required for soldering SD card socket. Place the socket properly and use tiny solder tip for each solder point. You can cut plastic bumps on the bottom side of the socket if you have any problem to place the socket correctly.

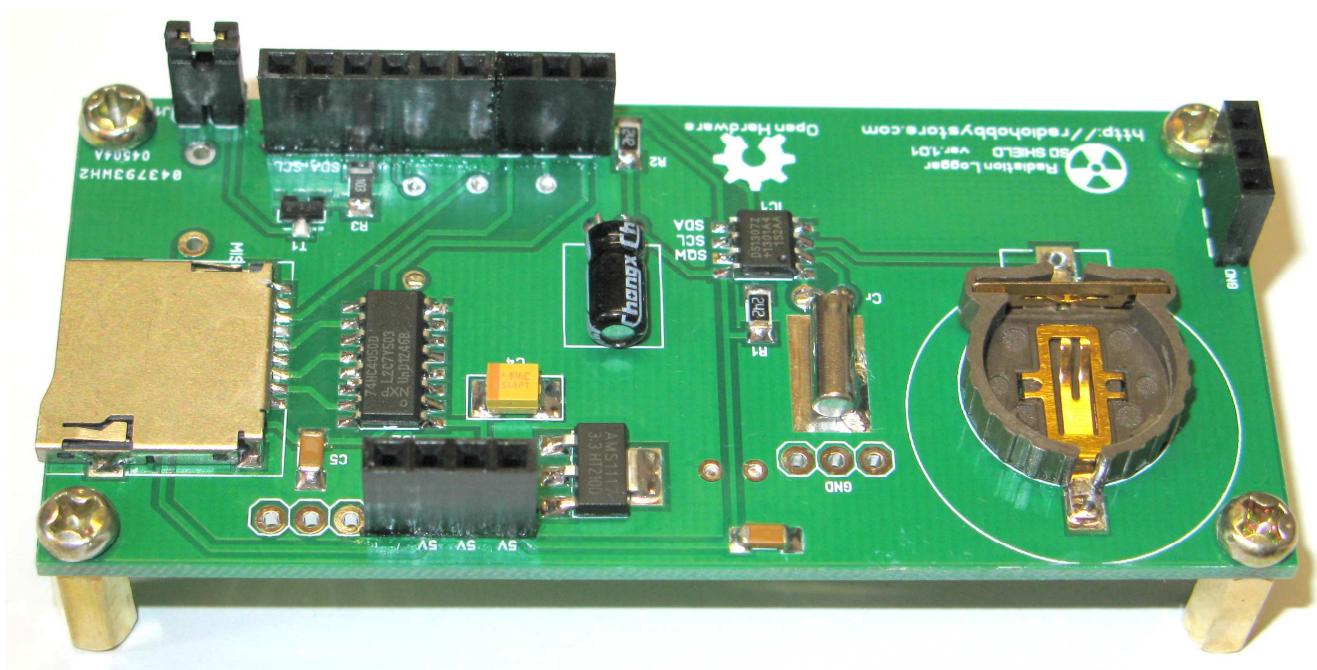
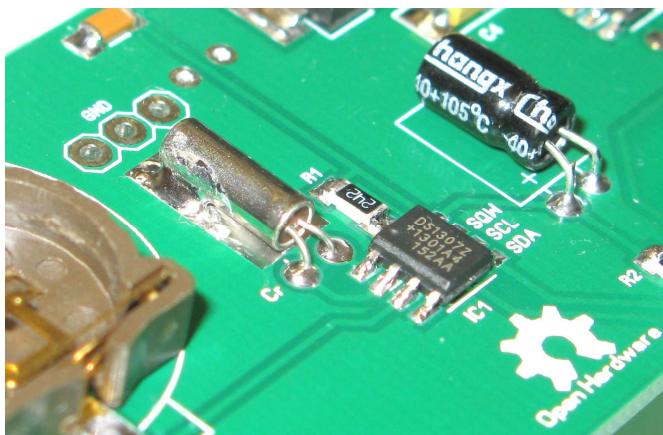


**Test each socket point it do not have unnecessary shorts to ground or to other socket pins!**

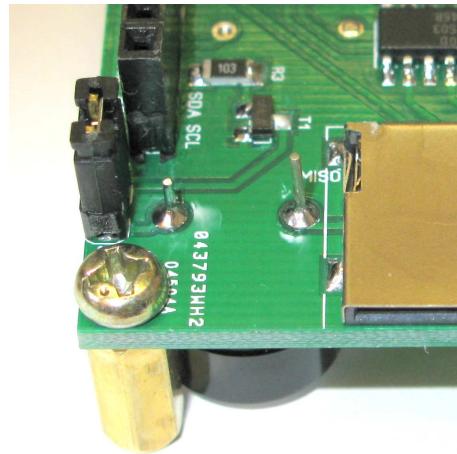




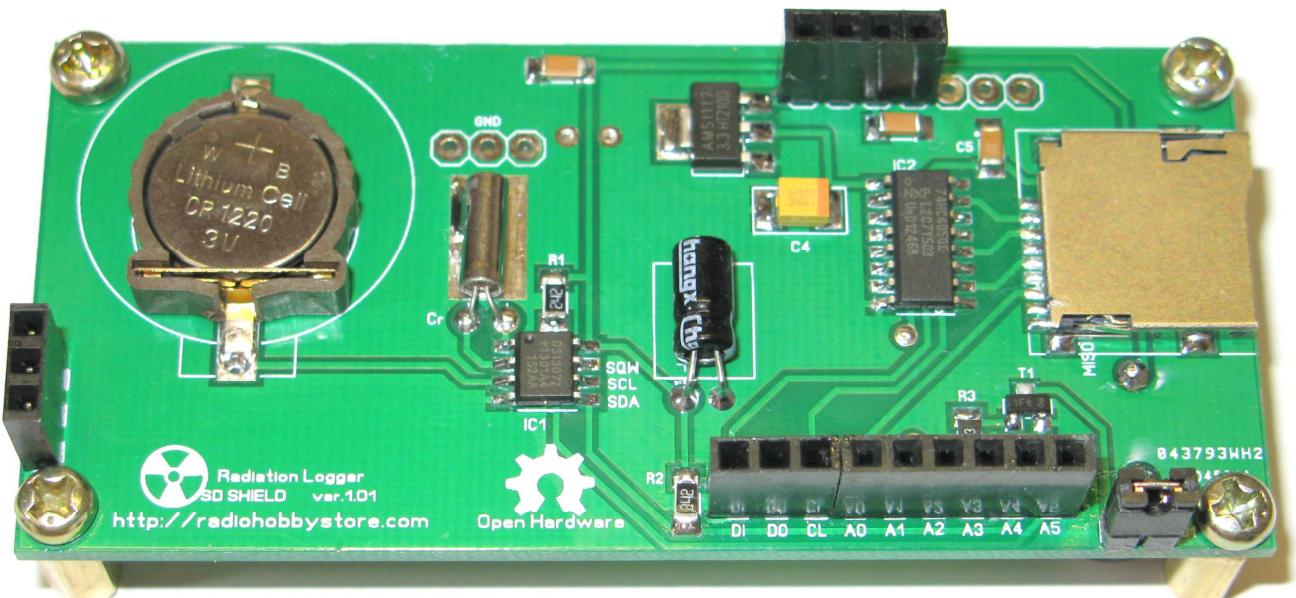
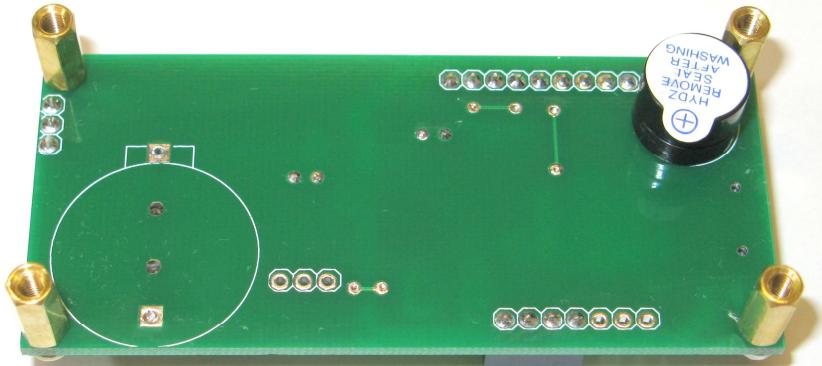
For RTC stability, crystal has to be soldered with point to ground plane. Do not overheat the crystal!

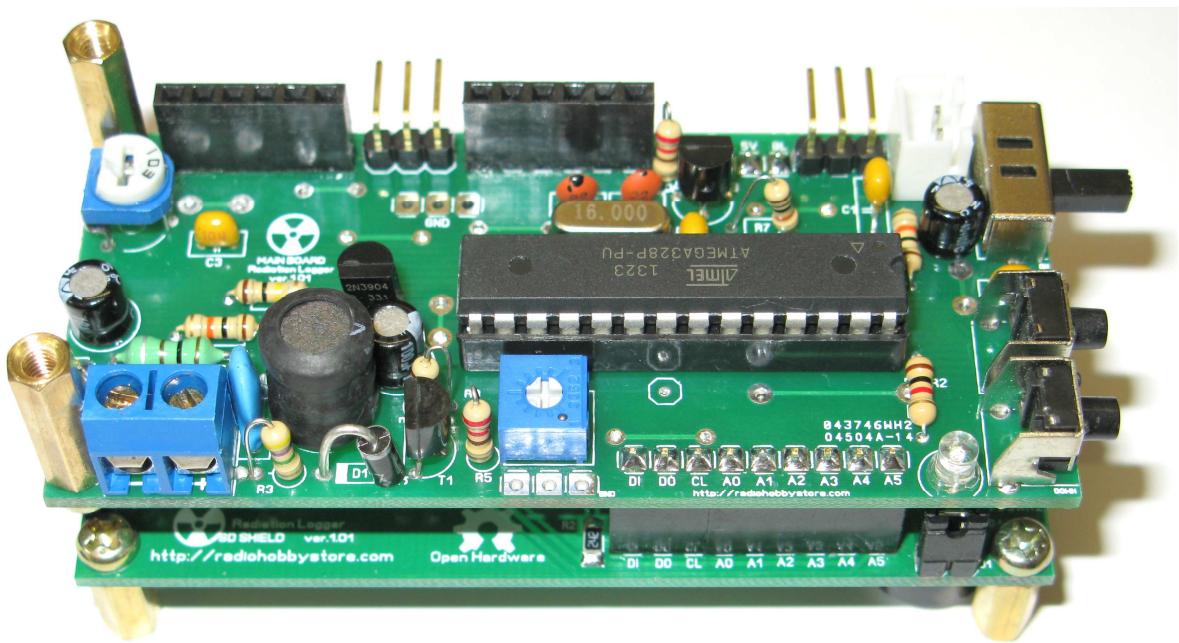


Solder all components except buzzer! Now you need to wash the board from both sides and. During wash the flux may create grease glue inside the buzzer and prevent it to work properly. That's why wash the pcb without buzzer, dry it for 30 minutes in ventilated area and only after that solder the buzzer.



Take note about buzzer polarity, the plus side toward SD socket. Remember to cut leads after soldering and remove paper sticker. You can connect external button to buzzer jumper pins in case you need mute switch. Install CR1220 battery for RTC. If you'll remove battery after sketch uploading, you'll need to re-upload date and time setup again, so keep this battery always in place. Now two boards are ready to be connected together for uploading the software.

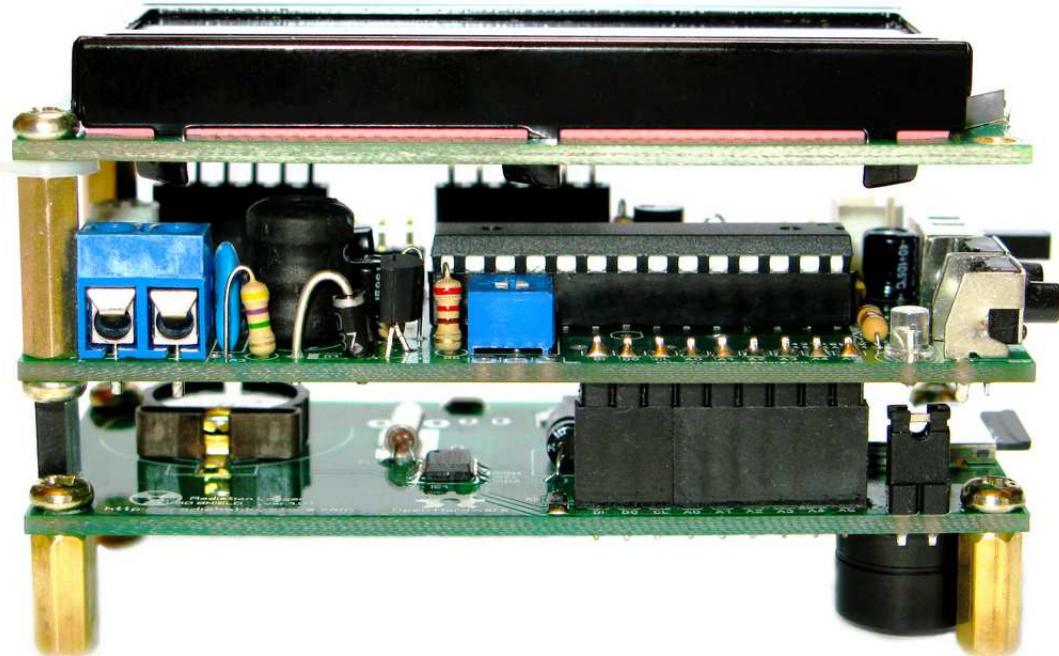




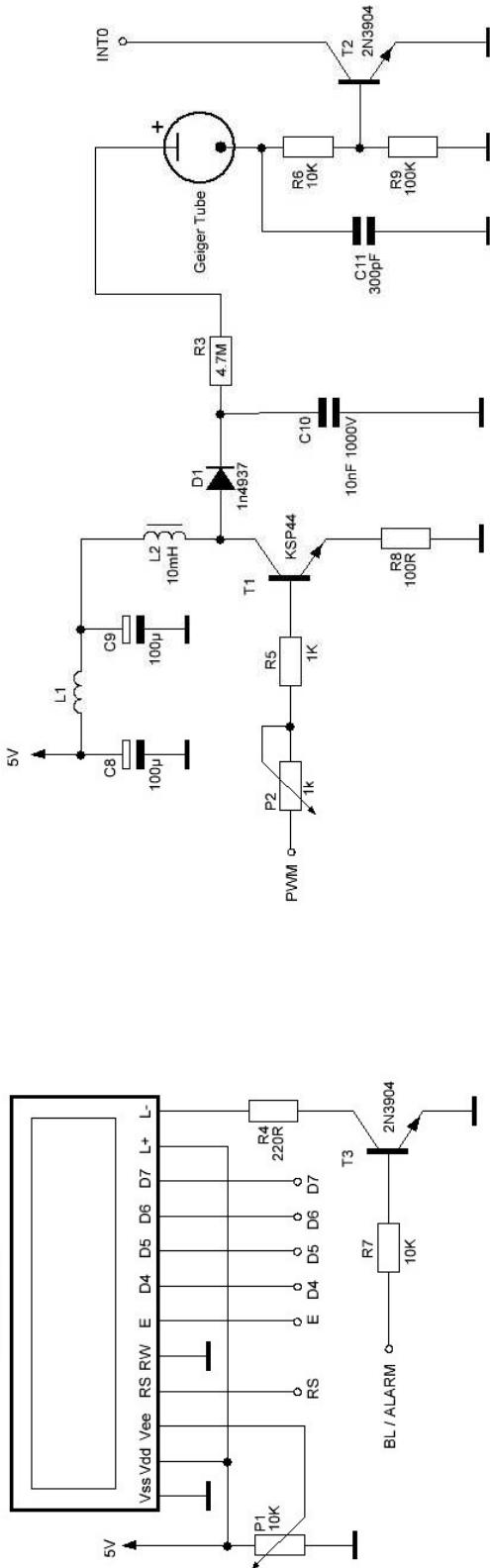
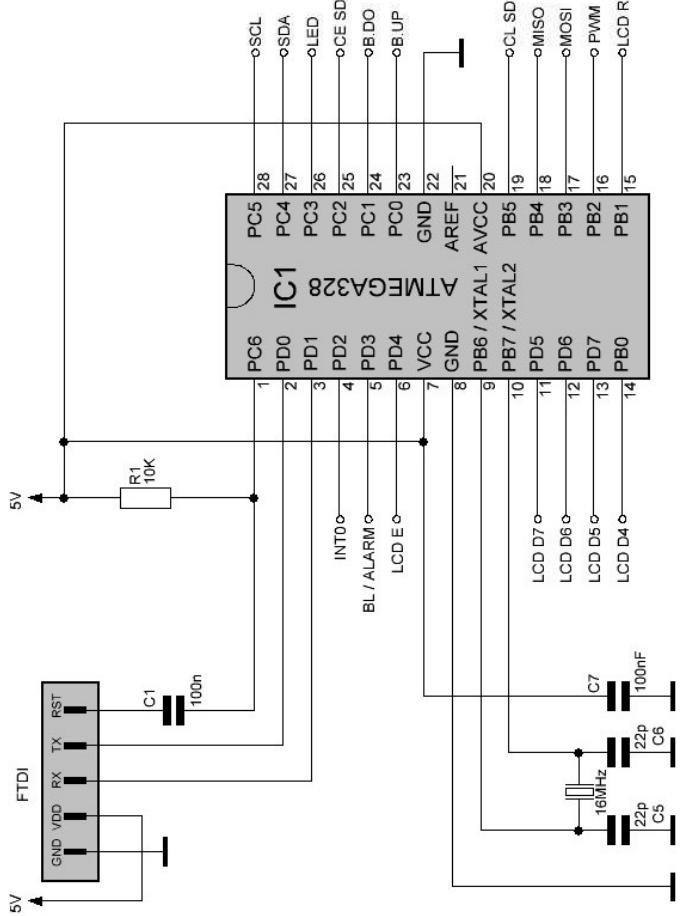
### Connecting all things together.

Insert main board pins into SD shield female headers. Install Atmega chip as shown on the picture above. It has pre-installed Arduino UNO bootloader, but you need to set RTC date and then upload geiger.ino sketch.

Why we not supply pre-programmed Atmega chip? Mostly because RTC IC need time and date setup. Without loading RTC setup sketch first, your date and time always will be shown as zeros or as erratic values. Firmware uploading is easy and made with Arduino IDE through USB-TTL module.



**Arduino Geiger Logger**  
**Main Board**  
ver. 1.00 20/12/2013  
<http://rhelectronics.net>

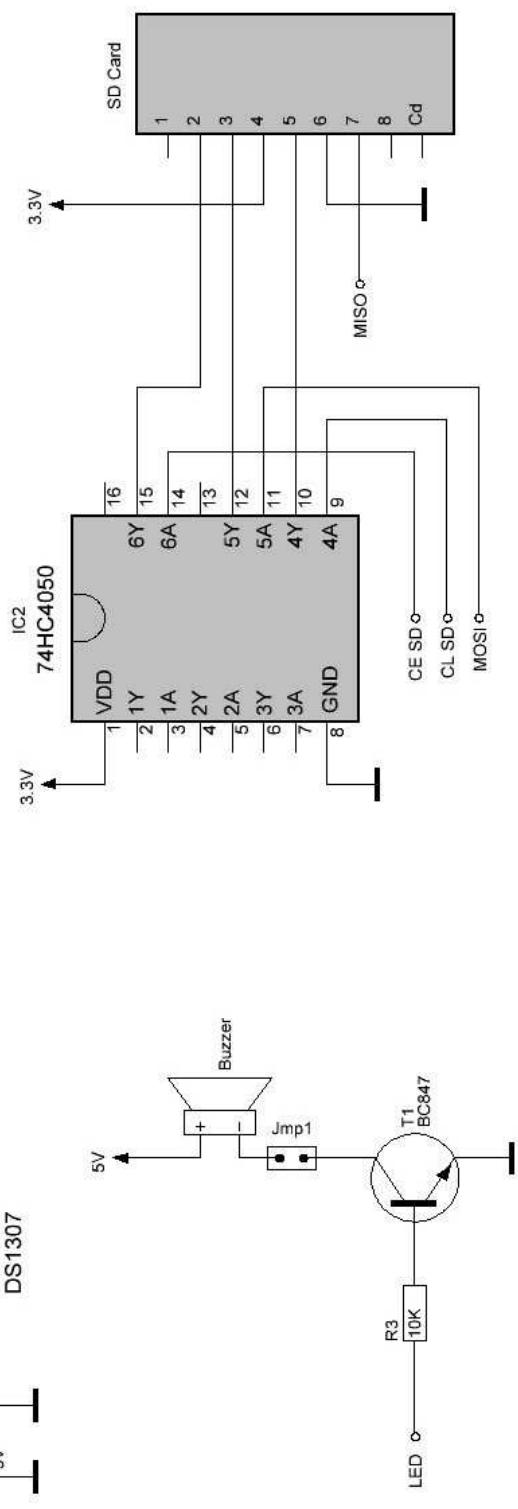
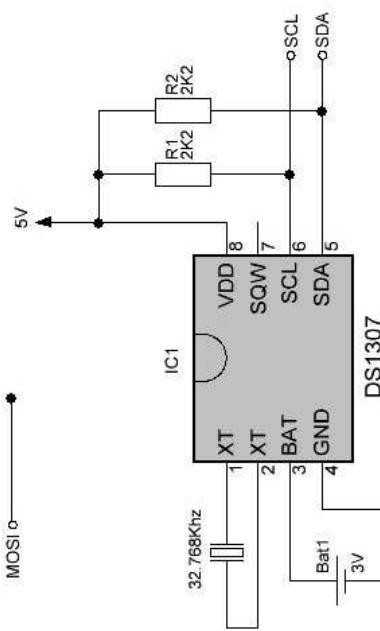
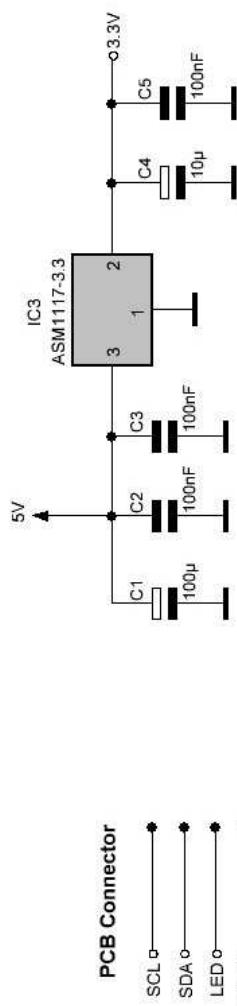


# Arduino Geiger Logger

SD Board

ver. 1.00 20/12/2013

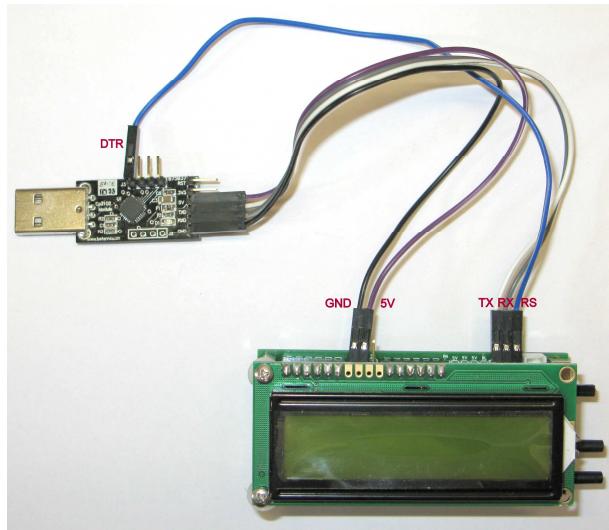
<http://rhelectronics.net>



Suggested that you already know how to upload sketch. Please do following steps:

1. Install Arduino IDE.
2. Install all libraries from supplied lib.zip
3. Install USB-TTL drivers for your adapter.
4. Open IDE, select Tools→Board→Arduino UNO. Select Tools→Programmer→AVRISP mkII. Set COM port where you connected your USB-TTL adapter.
5. Install RTC CR2110 battery.
6. Connect the board as picture below shows.
7. Open RTCsetup.ino, set your date and time and upload it into the board.
8. Open Geiger.ino, modify parameters if required and upload it into the board.
9. Disconnect the board from the computer and install micro SD card.

To upload sketch please connect USB-TTL adapter as shown on the picture. You can use a low cost CP2102 adapter with DTR pin. Atmega-328 comes with pre-loaded Arduino UNO bootloader that makes logger firmware updates extremely easy even for beginners through Arduino IDE. DTR pin should be connected to allow digital reset be processed for firmware upload.



#### **How to install Arduino IDE:**

Please download the latest IDE from <http://arduino.cc>. Navigate to "Download" section and select file for your OS. You can find general guides about how to use the IDE functions, for example Windows user can look here: <http://arduino.cc/en/Guide/Windows>. If you never used Arduino before, please take your time and read general guides first! You do not have to learn all, but the basic things.

#### **How to install libraries:**

Very important to use supplied libraries, otherwise you'll not be able to compile and upload the firmware. Please un-zip lib.zip into Arduino libraries folder or read here:

<http://arduino.cc/en/Guide/Libraries>

#### **Which USB-TTL adapter to use:**

Required module with Digital-Reset function (DTR pin), for example CP2102 is a cheap one can be found on <http://ebay.com>. Take note, some modules sold on ebay are fakes or missing DTR pin, please confirm with a seller about the model you buy.

### **What do I need to modify in RTCsetup.ino:**

You need to set your date and time and upload the sketch immediately. Please read program comments text. For example to set the 16:45:00 01/01/2014 I need to modify it like this:

```
RTC.set(DS1307_SEC,0);      //set the seconds  
RTC.set(DS1307_MIN,45);    //set the minutes  
RTC.set(DS1307_HR,16);     //set the hours  
RTC.set(DS1307_DOW,3);    //set the day of the week  
RTC.set(DS1307_DATE,1);   //set the date  
RTC.set(DS1307_MTH,1);    //set the month  
RTC.set(DS1307_YR,14);    //set the year
```

### **Which backup battery for RTC to use:**

CR1220 battery by default, but you also can use CR2032 (required different holder). After you upload RTC setup sketch do not remove the battery!

### **What do I need to modify in geiger.ino:**

Please read comments! Scroll down to load parameters section:

```
//----- load parameters-----//  
#define RAD_LOGGER true      // enable serial CPM logging to computer for "Radiation Logger"  
#define GPS false        // enable GPS on Serial Port  
.....
```

By default the sketch comes with GPS support turned OFF (false). If you do not use GPS module then turn ON Radiation Logger serial output

```
#define RAD_LOGGER true
```

### **What can I modify in geiger.ino?**

Mostly depend on your programming skills. Some users will make software redesign and some user will only change tube conversion factor. Please read comments in software. It will help you to identify code purpose for different parts of the program. You are free to modify the program for your consideration. Please remember when doing any changes that you are responsible for any modifications you make. The code provided as is and mostly for people who like to configure the hardware.

### **How to check and calibrate high voltage:**

The point here that you cannot measure high voltage with a regular voltmeter directly. It will overload the circuit because of low internal resistance of the voltmeter. Usually digital voltmeters has 1M or 10M internal resistance. That's why you need to use high impedance voltage divider. 1Giga ohm resistor is an ideal tool for this purpose (Mouser Part #: 588-SM106031007FE or similar). You need to connect your multimeter in series with 1Giga ohm resistor and calculate high voltage reading with following formula:

$$V_{out} = V_{read} * \frac{(R_{divider} + R_{voltmeter})}{R_{voltmeter}}$$

Where  $V_{out}$  is high voltage output,  $V_{read}$  is voltage you read on your voltmeter display,  $R_{divider}$  is 1000M ohm,  $R_{voltmeter}$  is your voltmeter internal resistance(1M or 10M). For example 404V after 1G divider with 10M multimeter will read as 4V.

$$404V = 4V * \frac{(1000+10)}{10}$$

In case you cannot find 1G resistor you can use 100M divider, connect 10x 10M 1% resistors in series.

**The optimum voltage range for SBM-20 tube is 380V - 480V, there is no difference for tube sensitivity. Because under high radiation load HV is dropping, it's recommended to trim HV on 450V-480V on background ambient radiation level.**

To calibrate your board high voltage, please power it with 5V and measure high voltage output when P2 is trimmed to middle position. If you read less than 380V or more than 425V then you'll need to adjust the software settings for high voltage. Open geiger.ino and find Configurator.h tab.

The following code is control PWM functions:

```
// PWM starting points, valid only for 400V without voltage multiplier!
#define PWM_FUL 100 // pwm settings for different battery levels
#define PWM_MID 105
#define PWM_LOW 110
#define PWN_FREQ 8 // PWM frequency setup
```

PWM\_FUL, PWM\_MID and PWM\_LOW it's the values you should change here. If you read lower than 380V try to increase PWM values with 5 points step. If you read higher than 425V try to decrease this numbers. Remember that all things connected together.

#### How to connect Adafruit GPS Module:

Before using GPS module you need to switch the software option in sketch and re-upload the sketch with GPS support turned ON.

```
//----- load parameters-----/
#define RAD_LOGGER false // enable serial CPM logging to computer for "Radiation Logger"
#define GPS true // enable GPS on Serial Port
.....
```

Connect GPS module to RX, TX pins. Adafruit module can be powered from the kit 5V pin. Take note, GPS will consume additional 20mA from the battery.

#### Other GPS modules compatibility:

Currently software support only Adafruit GPS module <http://www.adafruit.com/products/746>  
You can modify nmeaToGPS() function by yourself to initialize different GPS modules.

#### Why GPS not perform FIX:

GPS module requires open field condition. Set GPS antenna to sky direction.

#### How to prepare micro SD card:

You need to format the card before usage. Please follow Adafruit formatting guide here:

<http://learn.adafruit.com/adafruit-micro-sd-breakout-board-card-tutorial/format>

Download SD Formatter software [https://www.sdcard.org/downloads/formatter\\_4](https://www.sdcard.org/downloads/formatter_4)

**Do not format** with default Windows tool! When using card-reader insert or remove the card only then card-reader disconnected from USB! In the same way, insert or remove card only then the kit is powered off!

#### How to power the kit:

The kit's hardware needs 4.6V-5.4V to operate. There is several options for powering the kit, please consider the best for you.

1. 4x AA 1.2V Ni-MH rechargeable batteries, 1800mAh or higher.
2. 2x AA 1.5V Alkaline batteries with 5V step-up module.
3. 3.7V LiPO battery with 5V step-up module, 1000mAh or higher.
4. 5V DC regulated power supply for indoor.



**Powering the board with voltage over 5.50V or with wrong polarity may damage the components!**

Power consumption of the kit is about 20-22mA. LCD backlight require additional 8-10mA, but because the software use smart backlight control, most of the time LCD backlight is powered off for saving battery.

Battery indicator on LCD will inform you if capacity of the battery is half or low. Take note, battery indicator works correctly only with Ni-MH type. When using step-up modules battery indicator may be ignored or commented through software.

5V step-up module consumes additional current from the battery, depend on the module performance. **NCP1402** Pololu or Sparkfun 5V step-up modules are excellent for this kit, but also no-name eBay modules may work perfect.

### **Compatible Geiger Tubes:**

The kit has wide tubes compatibility. To convert CPM (count per minute) to uSv/h the software uses conversion factor. Usually it's known value from GM tube datasheet. Look at the factor settings code section and modify if other than SBM-20 tube is used.

```
// GM Tube setting
#define FACTOR_USV 0.0057 // Sieverts Conversion Factor, known value from tube datasheet
/*
SBM-20 0.0057
SBM-19 0.0021
SI-29BG 0.0082
SI-180G 0.0031
J305 0.0081
SBT11-A 0.0031
SBT-9 0.0117
*/
```

This values are not something absolute because each tube has about 10%-20% cpm tolerance. But for amateur purposes setting the default value for your tube is enough.

### **Logging to computer with “Radiation Logger”**

The kit allows you to log CPM values to your computer hard-disk with “Radiation Logger”. Download our freeware application for Windows here:

<http://www.rhelectronics.net/store/radiation-logger.html>

To connect the kit to your computer you'll need USB-TTL dongle adapter. You can use same adapter you have used for uploading the sketch or any other adapter with RX, TX, 5V and GND pins.

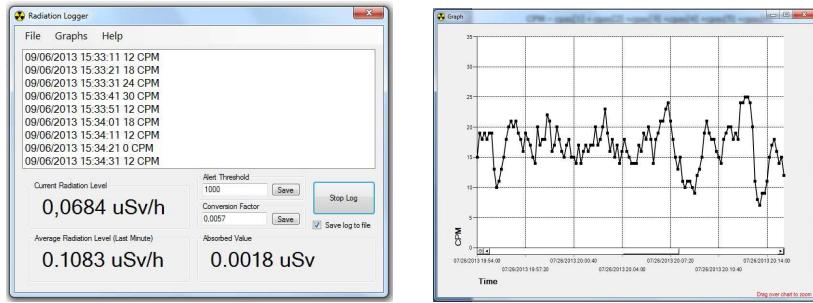
Use 9600 baud rate speed 8N1 settings for COM port. The logger will save csv log files on your harddrive with system time and CPM value. The files can be used for building graphs or comparing different measurements later.

With “Radiation Logger” you can log simultaneously to <http://radmon.org> and to your personal feed on <http://xively.com>. Connection settings are very simple.

```
//------------------ load parameters-----//
#define RAD_LOGGER true // enable serial CPM logging to computer for "Radiation Logger"
#define GPS false // enable GPS on Serial Port
.....
```



**When using “Radiation Logger”, the kit can draw power from the computer and it very important to set power switch to OFF to disconnect the batteries!**



## How to work with LOG.TXT and GPS data file:

Many thanks to great online tool available on <http://www.gpsvisualizer.com>. You can use LOG.TXT file to create google map with Geiger data you want to be presented. Here is the example. Part of the work done by auto and part of the work you need to do manually. Online GPS tool uses UTC time format from GPS data and not your local time, please take note when you edit CSV table.

Copy LOG.TXT file you want to work with from SD to your computer. When you open it with notepad you'll see that it contains different information, such as:

```

LOG22.TXT - Notepad
File Edit Format View Help
12/23/2013,06:08:25,16 CPM,63.24 uSv,4984 mV
$GPRMC,040903.040,V,,,,,0.00,0.00,231213,,,N*47
12/23/2013,06:09:25,23 CPM,63.24 uSv,4984 mV
$GPRMC,041003.000,A,3147.1361,N,03512.8326,E,2.59,284.41,231213,,,A*63
12/23/2013,06:10:25,17 CPM,63.25 uSv,5006 mV
$GPRMC,041103.000,A,3147.1471,N,03512.7736,E,2.73,287.76,231213,,,A*61
12/23/2013,06:11:25,19 CPM,63.25 uSv,5006 mV
$GPRMC,041203.000,A,3147.1617,N,03512.7185,E,2.99,291.30,231213,,,D*6A
12/23/2013,06:12:25,17 CPM,63.25 uSv,5006 mV
$GPRMC,041303.000,A,3147.1726,N,03512.6658,E,2.28,287.63,231213,,,D*65
12/23/2013,06:13:25,20 CPM,63.25 uSv,5006 mV
$GPRMC,041403.000,A,3147.1963,N,03512.6172,E,2.99,304.10,231213,,,D*66
12/23/2013,06:14:25,13 CPM,63.25 uSv,5006 mV
$GPRMC,041503.000,A,3147.2171,N,03512.5647,E,2.99,291.50,231213,,,D*64

```

Now we need to convert this file to something that gpsvisualizer website can work with. Browse to "Convert a GPS file to plain text" online tool [http://www.gpsvisualizer.com/convert?force\\_type=plain\\_text](http://www.gpsvisualizer.com/convert?force_type=plain_text)

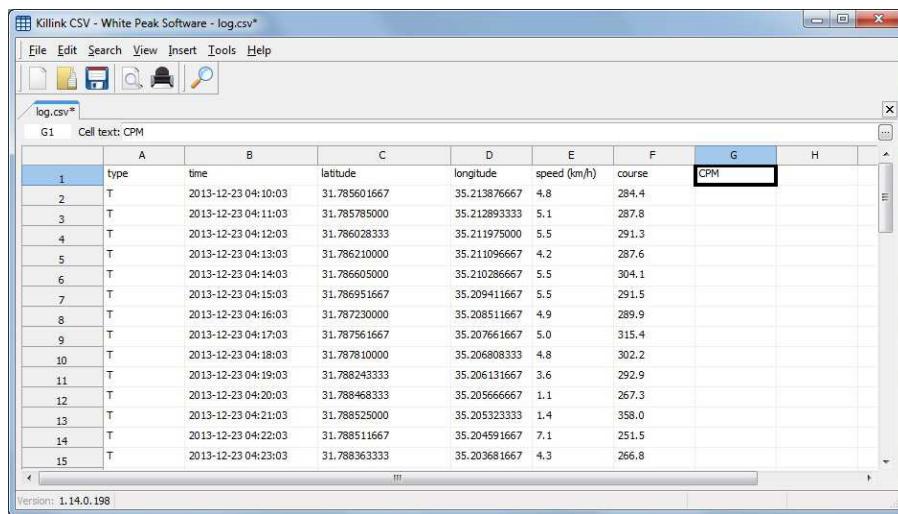
You'll get a similar screen when you can just copy-paste the table into your favorite CSV editor.

The contents of your file are also [displayed](#) in this box, if you'd rather cut and paste:

type	time	latitude	longitude	speed (km/h)	course	name
T	2013-12-23 04:10:03	31.785601667	35.213876667	4.8	284.4	LOG22
T	2013-12-23 04:11:03	31.785785000	35.212893333	5.1	287.8	
T	2013-12-23 04:12:03	31.786028333	35.211975000	5.5	291.3	
T	2013-12-23 04:13:03	31.786210000	35.211096667	4.2	287.6	
T	2013-12-23 04:14:03	31.786605000	35.210286667	5.5	304.1	
T	2013-12-23 04:15:03	31.786951667	35.209411667	5.5	291.5	
T	2013-12-23 04:16:03	31.787230000	35.208511667	4.9	289.9	
T	2013-12-23 04:17:03	31.787561667	35.207661667	5.0	315.4	
T	2013-12-23 04:18:03	31.787810000	35.206808333	4.8	302.2	
T	2013-12-23 04:19:03	31.788243333	35.206131667	3.6	292.9	
m	2013-12-23 04:20:03	31.788400000	35.205666667	1.1	267.3	

**Map this data:** [Google Maps](#), [Google Earth](#), [JPEG map](#), [SVG map](#), or [elevation profile](#) -- or go to the [map form](#) to set options

Add additional parameter you need, for example the CPM field.



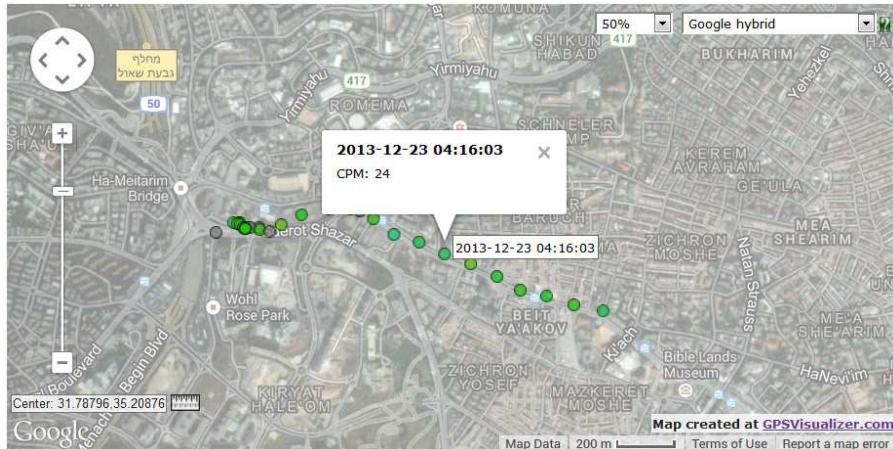
With hard manual editing enter CPM value for each data point to this file. Browse to online Data Plot tool here: [http://www.gpsvisualizer.com/map\\_input?form=data](http://www.gpsvisualizer.com/map_input?form=data)

This screenshot shows the "GPS Visualizer" website's "map input" form. It includes several sections for file upload, data entry, and map configuration.

- Upload your GPS data files here:** File #1: "log.csv" (highlighted with a red box).
- Or paste your data here:** "name,desc,latitude,longitude,n" (example data provided).
- General map parameters:** Output format: Google Maps (highlighted with a red box).
- Data point options:** Colorize using this field: "custom field" (highlighted with a red box), Custom colorization field: "CPM" (highlighted with a red box).
- Google Maps options:** Generate marker list: "No" (highlighted with a red box).
- Google Earth options:** Default icon: "Small circle" (highlighted with a red box).
- Contact information:** (empty section).

At the bottom, there are buttons for "Draw the map" (highlighted with a red box) and "Reset the form".

Here you can play with the settings to see the different maps you can get. And the result may look like:



By default SD Logger writes GPS data once per minute that great for walking with Geiger Counter. If you drive or fly, you may need faster logging. In this case please modify the sketch by yourself for saving LOG.TXT file with different time periods.

#### How to work with LOG.CSV file:

The file contains CPM readings points for every 10 seconds. You can view it through any csv file editor or with our "Radiation Logger" software. Press Graphs → Graph from log file and point to your LOG.CSV file "Radiation Logger" will build a graph with your saved data.

#### 600V High Voltage Mode

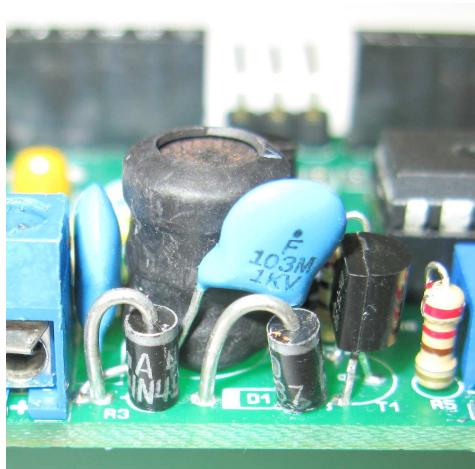
This mode requires ability to carry high voltage devices! Some Geiger tubes runs with anode voltage higher than 500V. Additional components for this mode are not included. You have to supply your own capacitors, diodes and load resistor.



**Any possible damage caused by using 600V mode is not covered by any warranty.**

If not carried properly, high voltage may damage kit components or lead to injury even the current is very low.

To perform this mode you'll need: 3pcs 1000V caps 2.2nF-10nF, use same capacity for all installed caps. 3pcs 1N4937 diodes and spare load resistor. Remove 4.7M load resistor from the main pcb. Install additional diode and capacitor as shown on the photo. The pcb has 2 additional small holes for this capacitor.



Then add external diode and capacitor. Load resistor should be installed on tube. This hardware mode adds voltage multiplier that allows to get 600V. Please modify PWM settings and upload new sketch before connecting multiplier! PWM values are experimental, start with 30 and increase it in small steps for getting 600V. Additional insulation required for external components! Use epoxy resin isolator or, at least, a heat shrink isolator.



**We do not recommend going over 600V for this kit!**

#### **ANY MODIFICATIONS ARE RESPONSIBILITY OF THE USER ONLY!**

Arduino is the open source platform for experiments. Geiger kit allows you to make changes in code, but you take full consideration of the results. RH Electronics provide you the tool with default sketch, but you decide what and how to modify the program. If you experience a trouble with your modifications it's all yours. RH Electronics support cannot help to everyone with the software and hardware modification. You can find tons of online information about writing program for Arduino IDE.

#### **WARRANTY DISCLAIMER**

This Geiger Counter kit is for EDUCATIONAL PURPOSES ONLY. You are responsible for your own safety during soldering or using this device! Any high voltage injury caused by this counter is responsibility or the buyer only. DO NOT BUY IF YOU DON'T KNOW HOW TO CARRY HIGH VOLTAGE DEVICES!

The designer and distributor of this kit is not responsible for radiation dose readings you can see, or not see, on the display. You are fully responsible for your safety and health in high radiation area. The counter dose readings cannot be used for making any decisions. Software and hardware of the kit provided AS IS, without any warranty for precision radiation measurements. When you buy the kit it means you agree with the disclaimer!

Copyrights:

RH Electronics

Alex Boguslavsky

<http://rhelectronics.net>

Copying project PCB design for commercial use is prohibited! You cannot produce this kit or PCB without my permission. Using this PCB design allowed only for non commercial goals for educational purpose or for home made amateur devices.