Environmental Monitor Assembly Manual

Revision 1 – released 2020-02-02

This kit is designed for someone who has basic experience with assembling electronics. This electronics kit requires knowledge of soldering, programming Arduino and a good grasp of basic electronics.

Please take your time – it will take approximately 3 hours to complete this kit. Ensure your work area is well lit (daylight preferred) and clean.
Assemble the parts in the order as stated in the instructions - read and understand each step before you perform each operation for the best chance of success.

The following tools and materials will be required to assemble the environmental monitor:
- A good quality soldering iron (25-40W) with a small tip (2-3 mm).
- Thin solder wire with no-clean flux. Do not use any flux or grease.
- A set of small screwdrivers.
- A small wire cutter for electronics.
- Long nose pliers.
- Tweezers.
- A multimeter.
- (Kapton) tape.
- A computer with the Arduino IDE and ESP32 Core for Arduino installed.
- A hair dryer (optional).
- A radioactive check source emitting beta or gamma radiation (optional).

Reading the entire manual before starting the project is highly advised and will help you comprehend the overall project.

This manual is written for Environmental Monitor software dated 2019-09-19.

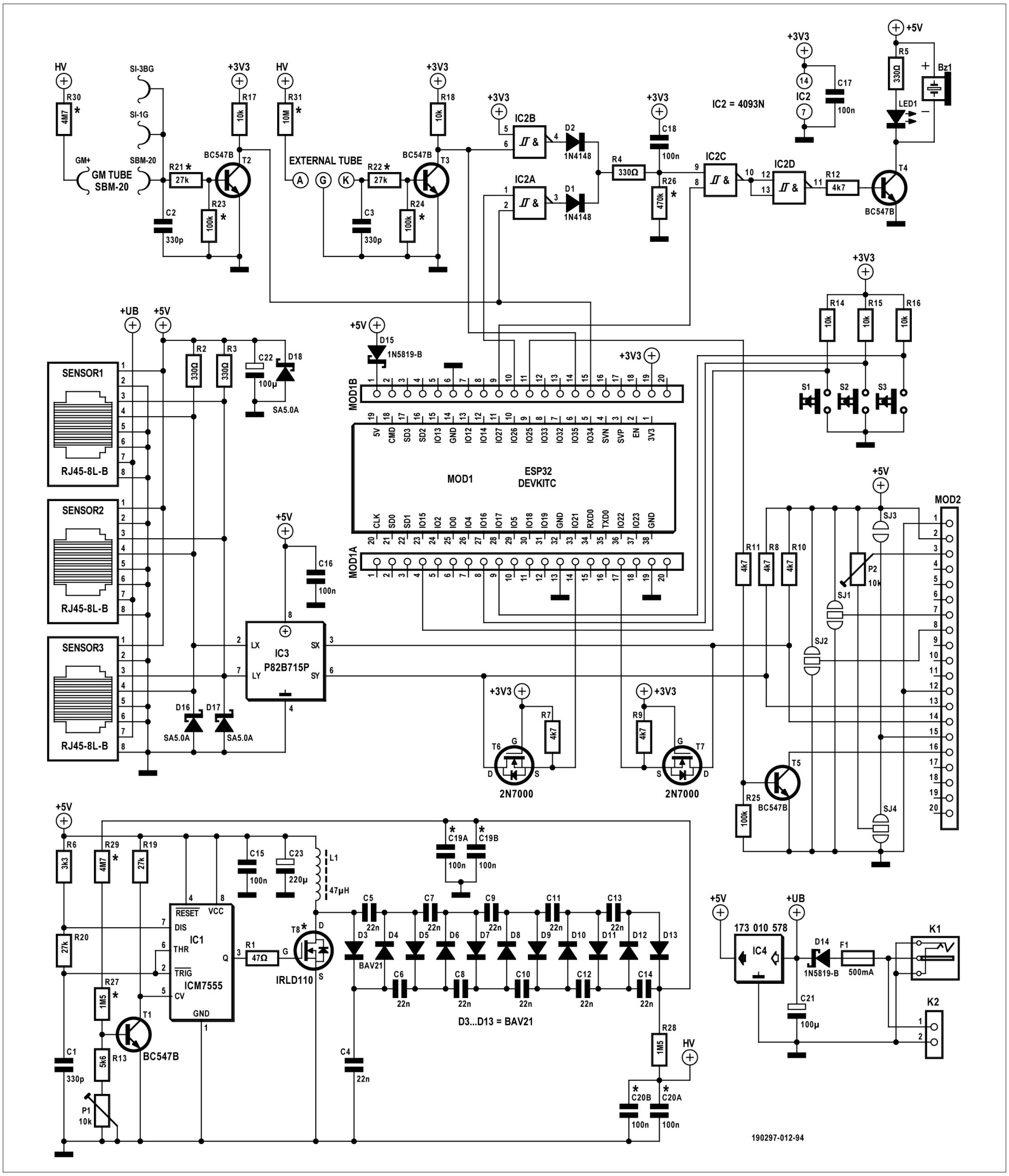
Caution
The high voltage (250 V – 1,1 kV) generated by this project is potentially hazardous. Do not touch the parts of the high voltage supply circuit when the device is operating. Despite the fact that the high voltage is not lethal, it still can provide a very painful shock.

If you intend to use the Environmental Monitor in a location where radioactive contamination is possible, it is strongly advised to wrap the entire device in a thin polyethylene bag. This allows radiation to pass through almost unhindered while preventing contamination of the instrument with radioactive particles, which are very difficult to remove.

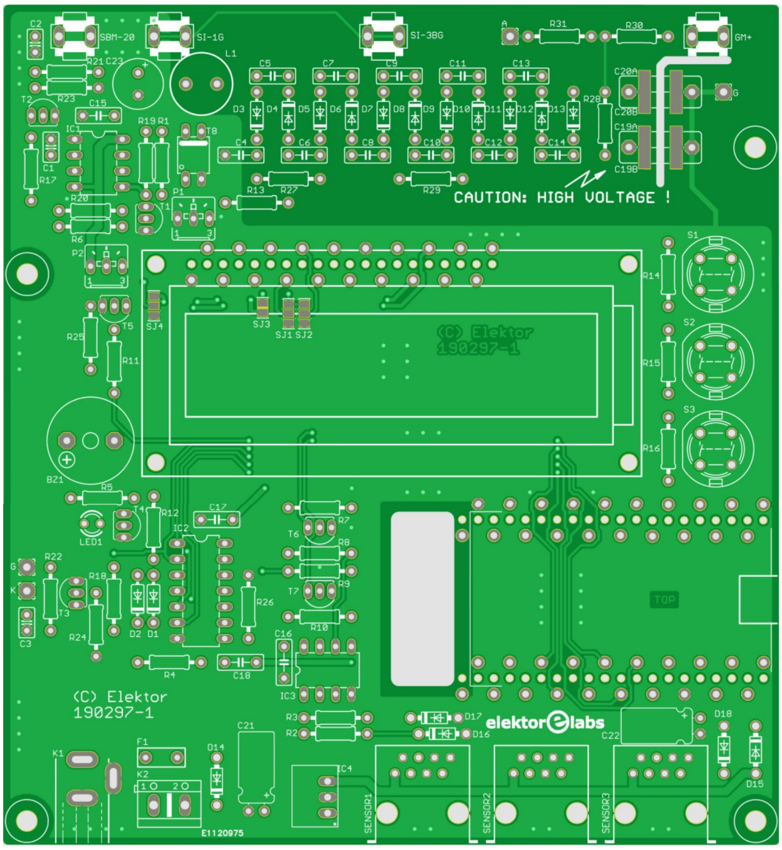
Disclaimer
All pictures are for illustration purposes only. Actual product may vary due to product enhancement.

Schematic and PCB Layout

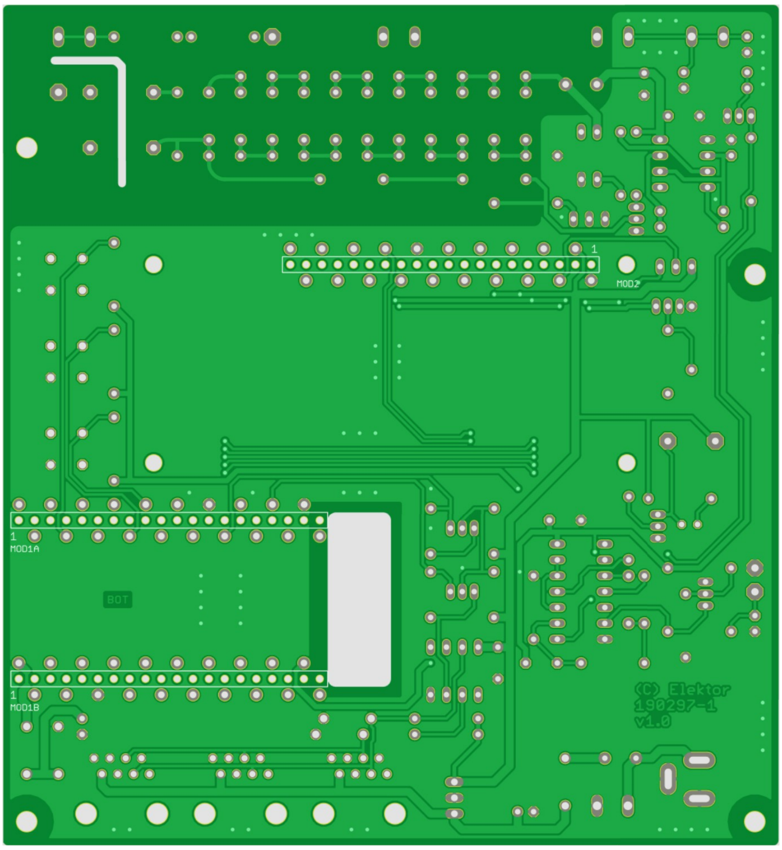
Circuit



Component Side



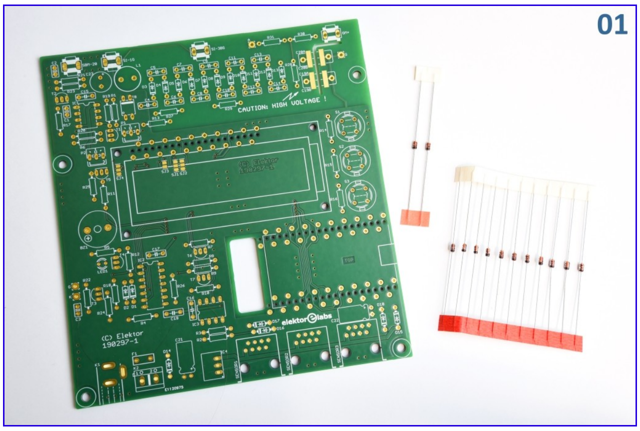
Solder Side



Assembling the PCB

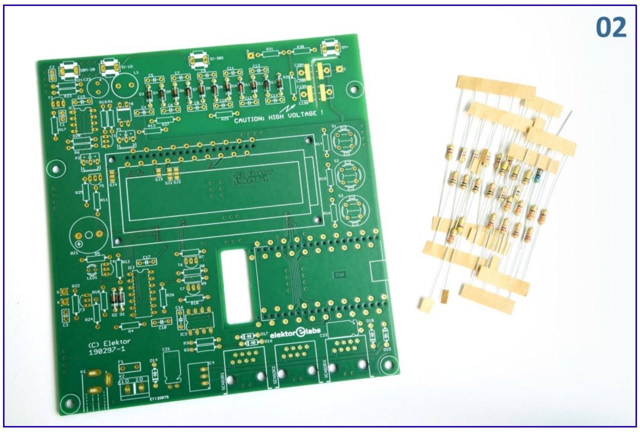
Step 01

Mount the components from low to high. Start with the small signal diodes D1, D2 (1N4148) and D3-D13 (BAV21). Watch the polarity!



Step 02

Mount the following resistors: R1 (47 Ω), R2-R5 (330 Ω), R6 (3,3 kΩ), R7-R12 (4,7 kΩ), R13 (5,6 kΩ 1%), R14-R18 (10 kΩ), R19-R22 (27 kΩ *), R23-R25 (100 kΩ *) and R26 (470 kΩ *). Reading the color bands of the 1% resistor and the high voltage resistors (mounted during the next step) can sometimes be difficult even if you have a very good color sight. If you are in doubt, check the value using a multimeter before mounting these resistors.

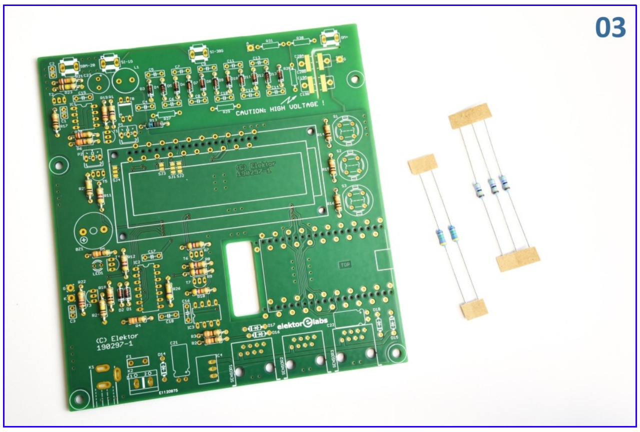


The resistor values indicated with “*” are supplied with the kit and are suitable when Russian SBM-20 Geiger-Müller tubes are used. When other types of tubes are used, the values of R21-R24 and R26 may need to be adapted for best performance.

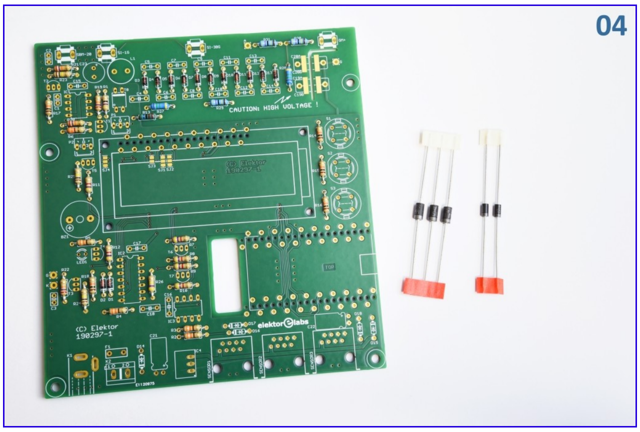
Step 03

Mount the high voltage resistors R27, R28 (1,5 MΩ), R29, R30 (4,7 MΩ) and R31 (10 MΩ, optional).

With the supplied resistors of 1,5 MΩ for R27 and 4,7 MΩ for R29, the output voltage of the high voltage power supply is adjustable between approximately 250 and 550 V. When using values of 5,6 MΩ for R27 and 6,8 MΩ for R29 (not supplied with the kit), the output voltage is adjustable between approximately 500 V and 1,1 kV. The recommended anode voltage for the SBM-20 Geiger-Müller tube is 400 V.



R30 (4,7 MΩ) is the anode resistor for Geiger-Müller tube 1 and is optimized for the SBM-20 tube. Unless you plan to connect a second tube, you can leave R31 open. The extra supplied 10 MΩ resistor is suited for small Geiger-Müller tubes from LND such as the LND 71217 and 712 tubes.

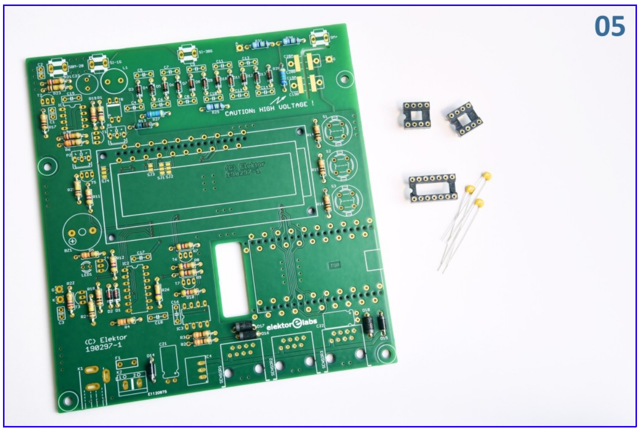


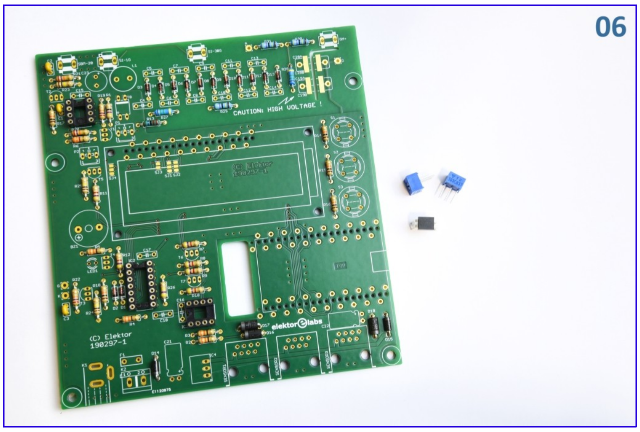
Step 04

Mount the Schottky diodes D14, D15 (1N5819) and the TVS diodes D16-D18 (SA5.0A). Watch the polarity! When mounting the TVS diodes, mount D16 and D18 first and make sure that D16 is not in the way for the RJ45 jack SENSOR1. Now bend the leads of D17 so it fits next to D16. This may be a bit of a tight fit.

Step 05

Mount the IC sockets for IC1, IC3 (8p) and IC2 (14p). Mount the ceramic capacitors C1-C3 (330 pF NP0 or C0G).





Step 06

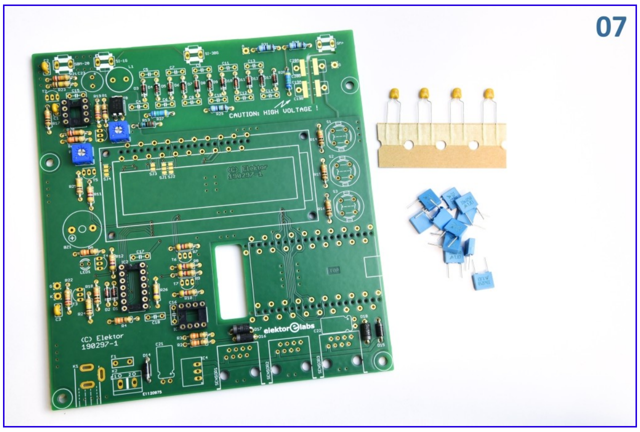
Mount the trim potentiometers P1 and P2 (10 kΩ). Make sure to mount them in the correct orientation. Turn trim potentiometer P1 all the way counterclockwise and set P2 to the middle position.

Mount the MOSFET transistor T8 (IRLD110), the side with the two leads connected together should match the dash marking on the PCB. MOSFET transistors are sensitive to static electricity. Make sure to discharge yourself or to take appropriate measures before handling the transistor.

For output voltages above 550 V, mount an IRFD220 transistor instead (not supplied with the kit).

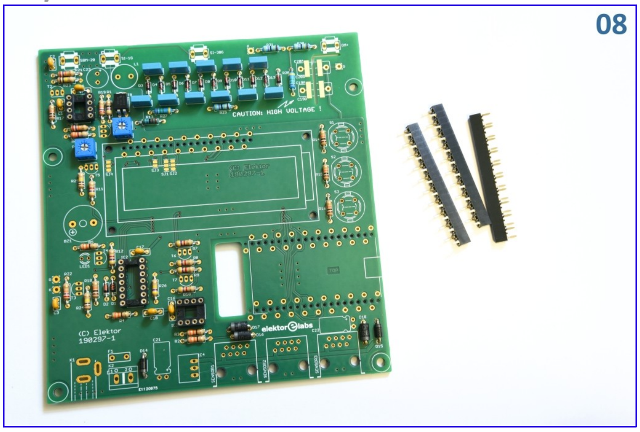
Step 07

Mount the MLCC capacitors C15-C18 (100 nF X7R) and the film capacitors C4-C14 (22 nF 250 V).

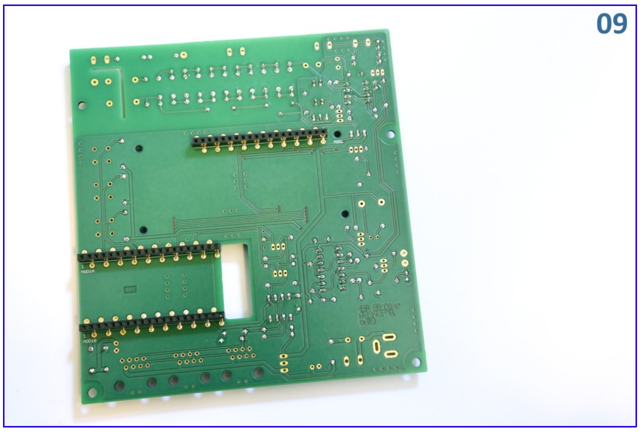


07

Steps 08-09

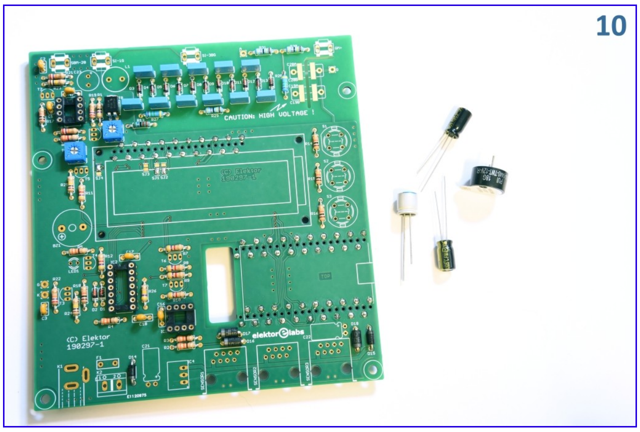


08



09

Mount the bottom entry socket headers for the LCD display module and the ESP32 DevKitC and solder them on the component side. You can trim the leads of the socket header which will be below the LCD display a little in order to prevent them to make contact with the metal frame of the LCD display module. As an alternative you can also put a bit of (kapton) tape over the solder joints.
Now is also the time to set the solder jumpers SJ1 to SJ4. These jumpers allow to accommodate for several types of LCD display modules and should be set according to the actual LCD display delivered with your kit. To make things easier, the kit contains a leaflet with information on how to set the jumpers. Please set the jumpers as instructed.

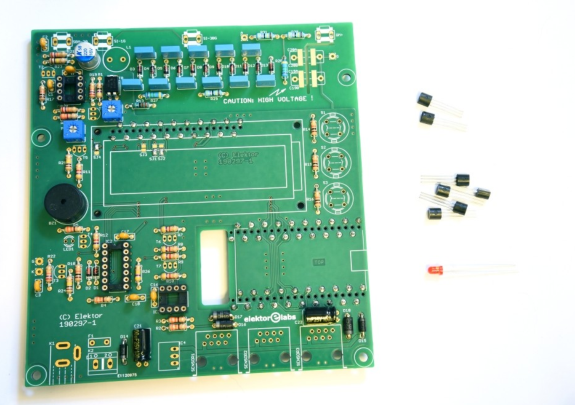


10

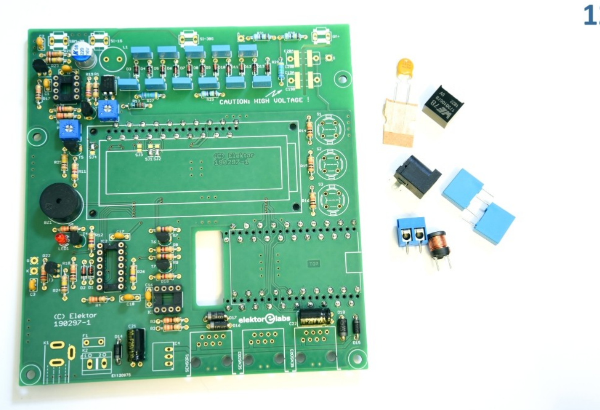
Step 10

Mount the electrolytic capacitors C21, C22 (100 µF 25 V) and C23 (220 µF 16 V). Make sure that C22 is not in the way of RJ45 jack SENSOR3. Watch the polarity!
Mount the buzzer BZ1 and remove the protecting seal. Watch the polarity!

Step 11


11

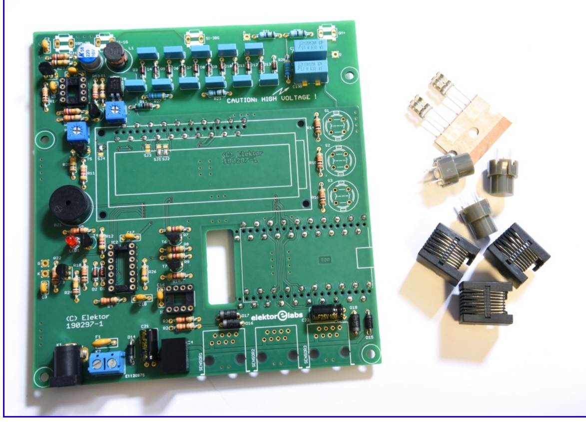
Mount the bipolar transistors T1-T5 (BC547B) and the MOSFET transistors T6, T7 (2N7000). Be careful about static electricity again when mounting the MOSFET transistors. Mount the transistors so that the top of their housings is flush with electrolytic capacitor C23.
Mount the 3 mm LED (LED1). Watch the polarity! The cathode (flat side on the housing of the LED) should point towards transistor T4. Mount the LED so it's flush with electrolytic capacitor C23 and be careful to not overheat it during soldering.
If desired, you can also mount an LED with a different color (not supplied with the kit).


12

Step 12

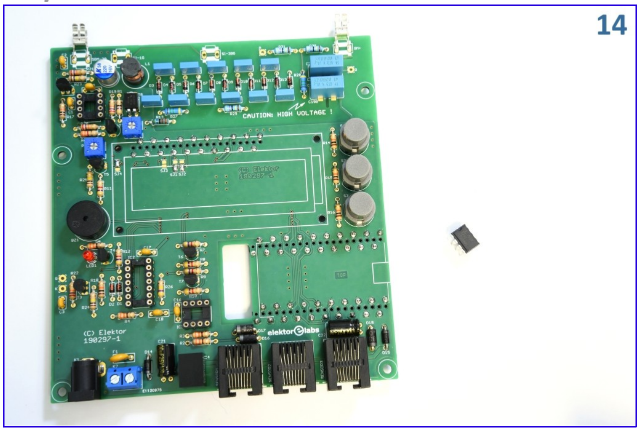
Mount the polyfuse F1 (Littlefuse 60R050XPR), the power inductor L1 (47 µH), the screw terminal block K2, the coaxial DC connector K1, the step-down regulator IC4 (Würth 173010578) and film capacitors C19A, C20A (100 nF 630 V).
For the higher output voltage range (550 V – 1,1 kV) you should mount C19B and C20B instead. These are large high voltage SMT capacitors (Kemet C2225C104KFRACTU, not supplied with the kit). As these capacitors are very sensitive to thermal shocks, you should not mount them manually with a soldering iron but use a reflow oven (before mounting any through-hole components) or a hot air station while slowly increasing the temperature. Exerting mechanical forces on the printed circuit board after mounting C19B and C20B should be avoided to the extent possible.

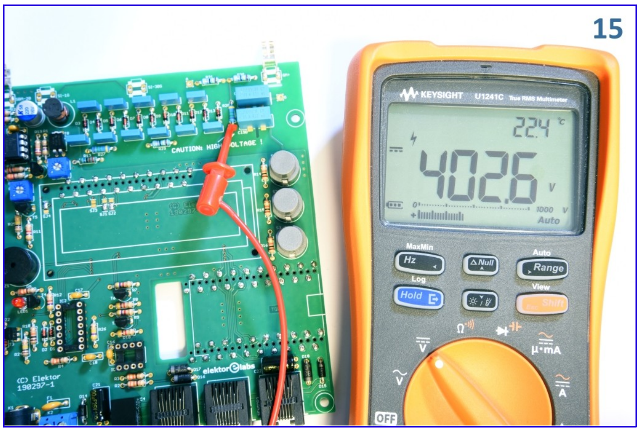
Step 13


13

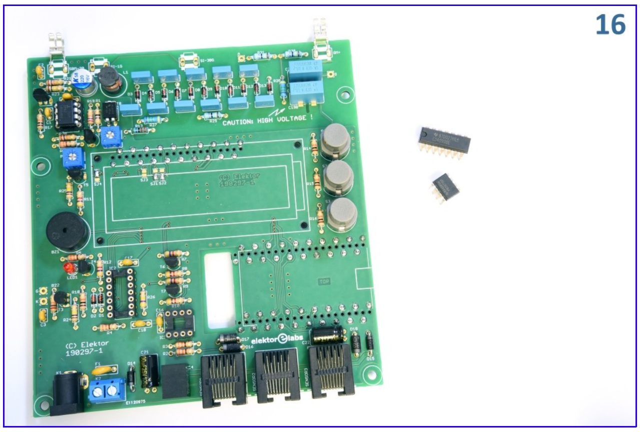
Mount the RJ45 jacks SENSOR1-SENSOR3 and push buttons S1-S3. Be careful not to overheat the push buttons when soldering them to the board.
If you plan to use one of the Russian Geiger-Müller tubes like the SBM-20, the SI-1G or the SI-3BG, you can mount the fuse clip holders to the corresponding solder pads.
Bend the leads of the fuse clip at an angle of 90 degrees and mount them as in the pictures. Widen the fuse clips a bit by bending them carefully to allow the Geiger-Müller tube to fit more easily.
Please note that we don't recommend using the SI-1G or SI-3BG tubes except for very specific purposes and that we haven't tested the circuit with these tubes.

Steps 14-15





Put IC1, an ICM7555 into its socket and power up the board using a 9-12 V DC power supply.
Check the 5 V supply voltage with a multimeter. The easiest way to measure this is to measure between pin 4 and pin 8 of the socket for IC3.
Now measure the voltage across capacitor C19 and adjust it to approximately 400 V (when using an SBM-20 Geiger-Müller tube) with trim potentiometer P1.

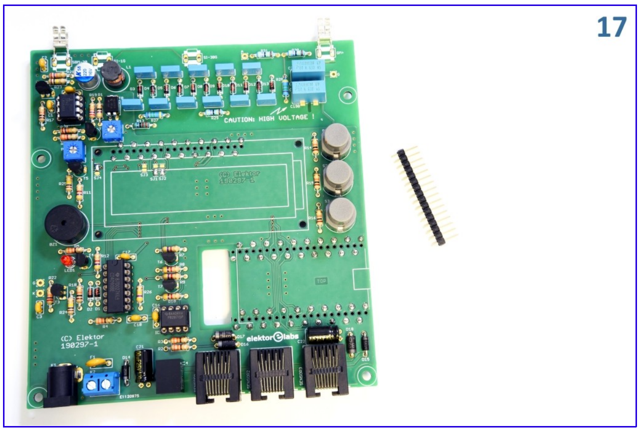


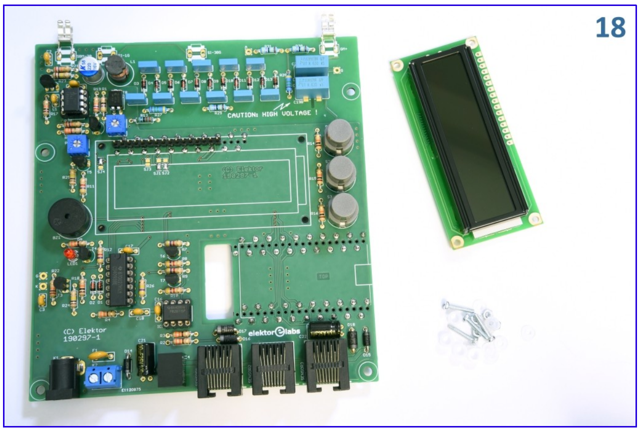
Step 16

Remove the power and install IC2 (CD4093N) and IC3 (P82B715P) in their socket.

Step 17

Push the 16-way pinheader into the bottom entry header for the LCD display from the component side.
The bottom entry socket headers delivered with this kit are 20-way as these were easily available. This means that not all contacts are used. The 16-way pin header should be inserted with pin 1 at the leftmost position of the bottom entry socket.

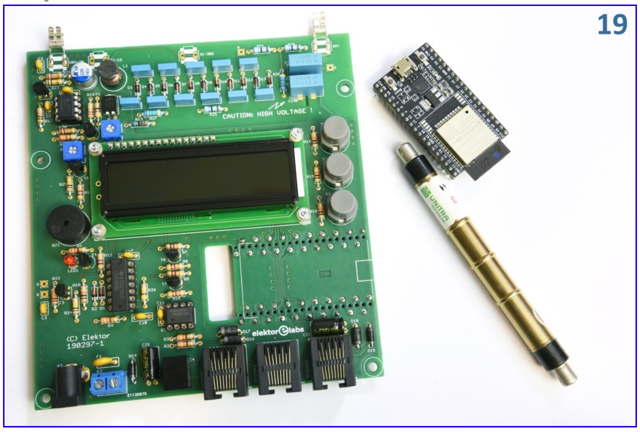


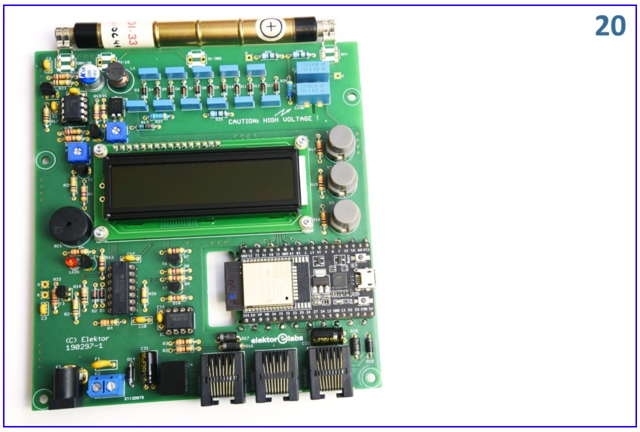


Step 18

Take four M2X10 screws and slide an M2 nylon washer over them. Turn the PCB upside down and guide them from the solder side through the mounting holes for the LCD display module. Use four pieces of tape to hold the screws temporarily in place and flip the PCB again.
Slide four nylon 3 mm spacers over the protruding screws and carefully lower the LCD display module over the screws and the pinheader until it is in place. Now secure the LCD display module with the remaining M2 washers and four M2 nuts.
Finally solder the 16-way pin header to the LCD display module. It is recommended to cover the display area of the LCD display during this operation to avoid contamination of the glass surface with solder flux spatters.
Remove the pieces of tape from the solder side of the PCB.

Steps 19-20





Load the firmware into the ESP32 DevKitC module using a PC with the Arduino environment and ESP32 Core for Arduino installed. You can do this before or after installing the ESP32 DevKitC module onto the board of the Environmental Monitor.
If the ESP32 DevKitC is installed onto the PCB of the Environmental Monitor, please make sure that the circuit gets external power from a 9 – 12 V DC supply. Otherwise the entire circuit will be powered from the computer's USB port through the ESP32 DevKitC module which is not optimal.

When installing the ESP32 DevKitC module, make sure the WiFi antenna is facing the milled rectagular hole in the PCB and that it is aligned correctly as it has 19-way pin header connectors while the bottom entry sockets have 20 contacts. Mount the ESP32 DevKitC so that the contacts of the bottom entry sockets closest to the rectangular hole are not used.

Finally connect the Geiger-Müller tube. If you are using a Russian SBM-20, you can simply push the tube into the fuse clips on top of the PCB. Watch the polarity! If you are unsure, you can use a multimeter to find the cathode of the Geiger-Müller tube as it is usually connected to the metal wall of the tube. Installing the Geiger-Müller tube with the wrong polarity does not damage it, but it will not detect ionizing radiation very well. Never, ever, solder directly to a Geiger-Müller tube as the heat may damage the glass seals of the tube!

Power up the Environmental Monitor and adjust the contrast of the LCD display using trim potentiometer P2 until text becomes visible. You may hear the buzzer beeping occasionally and see the LED flashing. If not, you can turn this on via the push button menu. Congratulations, you are now detecting background radiation. If you wish you can now also test with a check source if you have one.

You will find more information on buying Russian Geiger-Müller tubes and check sources at the end of this assembly manual, so stay tuned!

Assembling the Enclosure

Remove the protective foil from the acrylic bottom panel and the side panels of the enclosure.

Attach the four black 8 mm standoffs to the bottom panel using the M3X6 countersunk machine screws.

If you wish you can stick four rubber bumpers to the underside of the bottom panel.

Position the assembled PCB of the Environmental Monitor on the 8 mm standoffs and put the side panel with the recesses for the RJ45 jacks in place.

Secure the PCB to the standoffs using the plastic M3 washers and four 11 mm metal standoffs. Hand-tighten the standoffs and put the remaining side panels into place. Remove the protective foil from the LCD module if it is still present. Remove the protective film from both top panels and position the smallest top panel on top of the Environmental Monitor PCB so the LCD module and the push buttons protrude. Make sure there isn't any dust trapped between the LCD module and the top panel. Now put the second top panel on top and secure it to the metal standoffs using four M3X10 machine screws. Check again for annoying dust particles in the LCD display area. The Environmental Monitor is now ready for use.

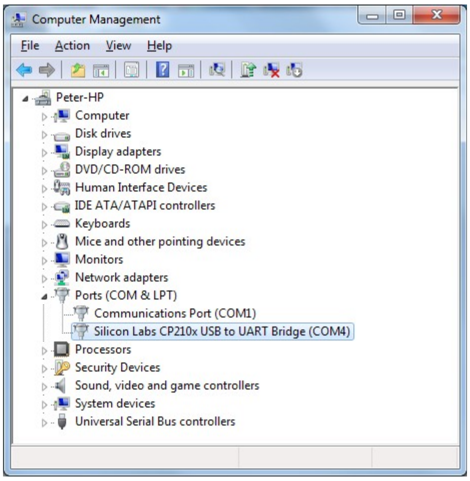
Setting up the Software

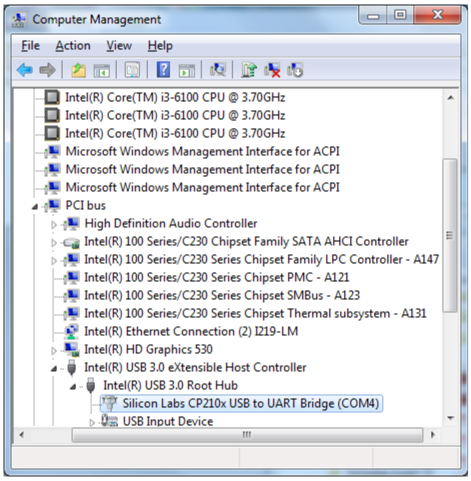
This section is for the most part aimed at using the Arduino IDE in Windows 7. You should have no problem when using another version of Windows or any other operating system that's supported by the Arduino IDE.

Serial Interface

Power up the Environmental Monitor and connect the ESP32-devkitC with a PC using a USB cable. The ESP32-devkitC incorporates a USB-to-serial function that enables the PC to access the board as a serial interface.

If you're using Linux, the system will create an entry in the device directory structure e.g. **/dev/ttyUSB0**. Run command **lsusb** to obtain a list of all connected USB devices.





If you're using Windows, the operating system will assign a COM-port. A serial port is typically named **COM<x>** where **<x>** is a number between 1 and 256. You can observe the COM-port in the device manager. When the devices are displayed by type, the COM-port appears under the “Ports (COM & LPT)” branch. You can select to view devices by connection which renders a tree structure.

Environmental Monitor Software

The Environmental Monitor software is an Arduino sketch. We used **Arduino IDE v1.8.9** for developing and testing the software. The source code is made up of .ino files. You find many useful remarks and comment blocks in these source files.

Open **envmon.ino** in the Arduino IDE to open up the Sketch. The IDE shows all files in tabs, starting with the main file and followed by the other files in the folder sorted alphabetically.

The sketch required the following libraries: **Wire** (for access to I2C), **EEPROM**, **WiFi**, and **Time**. If any of these libraries is missing, open the library manager (menu Sketch -> Include Library -> Manage Libraries...) and install the missing pieces. If multiple versions of a library are available, use the latest one.

Before you start building the sources, you've to set up the IDE for use with the ESP32-devkitC module. In the Tools menu, select board type “ESP32 Dev Module” and select the COM-port assigned to your Environmental Monitor.

Now you can build and upload the software. In the Arduino IDE, this process is simply called “Upload”. Select menu Sketch -> Upload, or press CTRL-U to proceed.

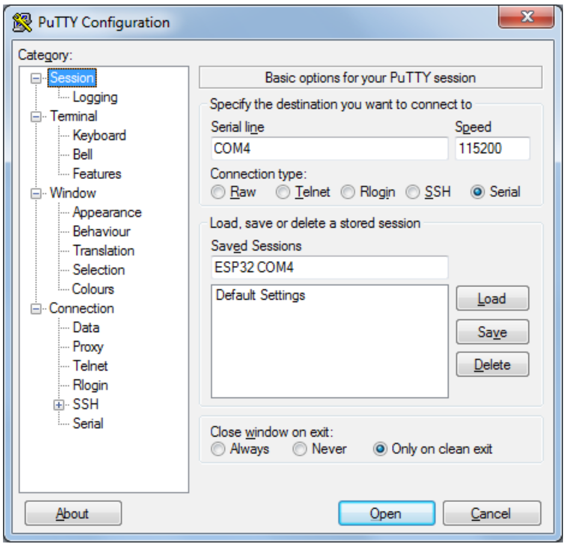
After the IDE has successfully uploaded the binary to the ESP32-devkitC, the Environmental Monitor sends text to the serial interface. You can observe this information in the IDE's serial monitor. Select menu Tools -> Serial Monitor, or press CTRL-SHIFT-M to open up the serial monitor. Set the baudrate to 115200 and select something different from "no line ending".

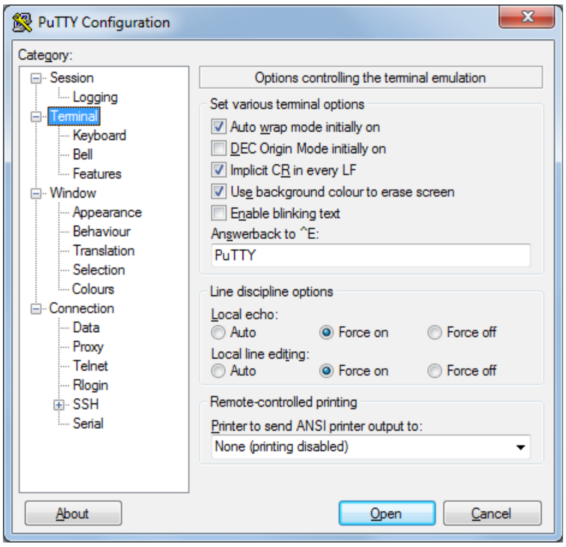
IMPORTANT! Each time you open the serial interface, the board will reset. This behavior is specified by Arduino and occurs with all Arduino-compatible boards, independent of operating system and serial monitor program (Arduino IDE's serial monitor, PuTTY, netcat, ...). So beware, when you change settings, then close and reopen the serial interface before writing the settings to non-volatile memory (a.k.a. EEPROM), the changes will be lost.

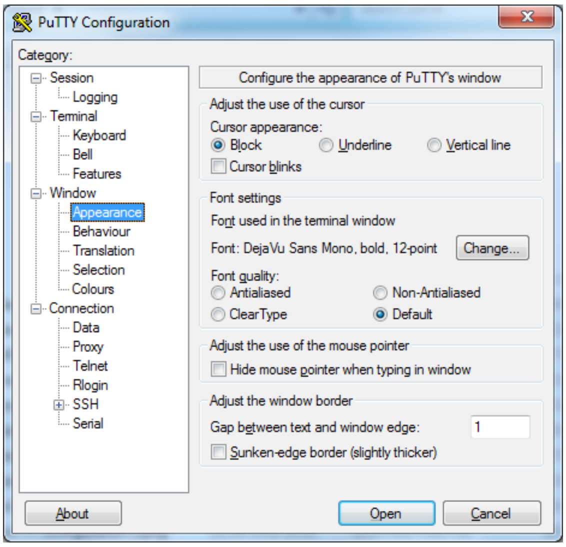
You can send commands to the Environmental Monitor from the serial monitor. For example, command **s d** (a short version of **settings dump**) reports a list of all current settings.

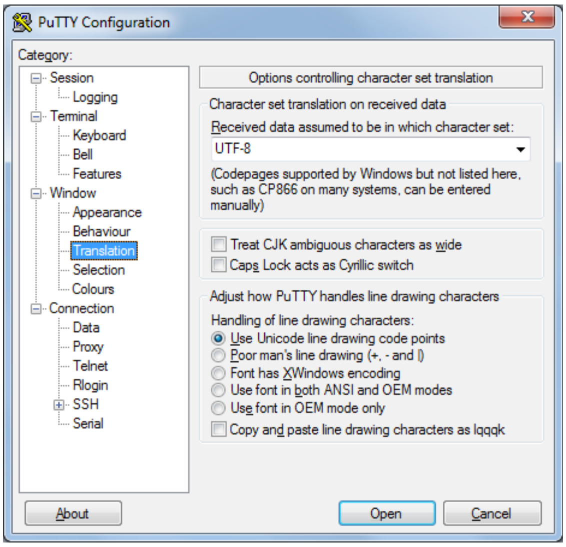
At this point, no settings are set, unless you're ESP32-devkitC was already set up for the Environmental Monitor. To fill in the settings, you've to send a series of commands. You can manually type each command in the serial monitor. This will take time, so let's use a better method.

You can put the commands in a text file and send its contents to the serial port. Since the serial monitor in the Arduino IDE isn't capable of sending a file or pasting multiple lines from the clipboard, we're going to use another serial monitor. A good choice is PuTTY in Windows and Linux. Another good choice is netcat in Linux.





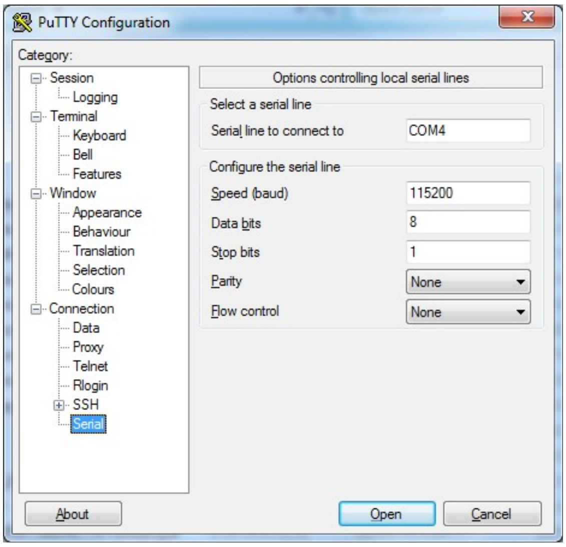




The most important PuTTY settings are:

- Select the COM-port assigned to the Environmental Monitor.
- Select baudrate 115200, 8 data bits, 1 stop bit, no parity, no flow control.
- Turn on implicit CR in every LF.
- Set local echo to “Force on”.
- Set local line editing to “Force on”.
- If your screen has a high resolution, you may want to select a larger font.
- Use the UTF-8 character set.

You can save these settings in a named session.



Click “Open” to make a connection with the board. Once connected, you can send commands to the Environmental Monitor, like **w d** (short for **wifi dump**).

Create a text file (e.g. settings.txt) that contains the necessary commands for setting up the Environmental Monitor. Many schemes are possible. You can find a list of commands in a later section.

When the text file is done, you can transfer it to the serial port in the serial monitor. In the case of PuTTY, select all text in the editor and copy it to the clipboard. CTRL-C will do the trick. In PuTTY, press the right mouse button to paste the text from the clipboard. The settings will take effect as soon as the Environmental Monitor receives them.

We have to store the settings in non-volatile memory to preserve them when the Environmental Monitor is powered down or reset. For convenience, you can include command **settings write** in your text file so all settings are immediately written to non-volatile storage.

The Environmental Monitor software allows a network client to connect using port 5010. We can use PuTTY, netcat, or similar terminal program to accomplish this. Once the terminal program has established a connection with the Environmental Monitor, you can send commands the same way you do with the serial monitor. Note that unlike opening the serial interface, establishing a connection over the wireless network won't reset the ESP32-devkitC module.

Besides using a PC, you can use your smartphone to send commands to the Environmental Monitor over the wireless network.

Here are the contents of an example settings file. You can copy the text from this document in your own editor and adjust the settings to your liking.

```
wifi set ssid "your ssid"
wifi set password "your password"
wifi set local 192.168.1.141
wifi set subnet 255.255.255.0
wifi set gateway 192.168.1.254
wifi set dns 1 192.168.1.254
wifi set dns 2 0.0.0.0
wifi set method static
wifi set hostname "your hostname"

serial set echocmd yes

cs set echocmd no

ntp set hostname "pool.ntp.org"
ntp set addr 0.0.0.0
ntp set method hostname

clock set selectadjust 2
clock set adjust 1 + 01 00 00
clock set adjust 2 + 02 00 00
clock set dateformat dmy
clock set hourformat 24
clock set hourlz yes
```

```
main set wificonnect yes
main set ntprequest yes
main set ntprefresh 01 00 00
main set cmdsrvstart yes

gm 1 set attached yes
gm 1 set accperiod 10
gm 1 set deadtime 190
gm 1 set cnvfactor 3931
gm 1 set threshold 5000
gm 2 set attached no

thingspeak set gm 1 writekey  "your key"
thingspeak set gm 1 fieldname "your fieldname"
thingspeak set gm 2 writekey  "your key"
thingspeak set gm 2 fieldname "your fieldname"

opensensemap set gm 1 senseboxid "your sensebox id"
opensensemap set gm 1 sensorid   "your sense id"
opensensemap set gm 2 senseboxid "your sensebox id"
opensensemap set gm 2 sensorid   "your sense id"

settings write
```

Commands

The Environmental Monitor software accepts an extensive list of commands. There are several groups of commands, each starting with a specific keyword e.g. **wifi**. Many commands can be abbreviated. For example, **w set m st** is short for **wifi set method static**.

Commands that change a setting have a 2nd keyword called **set** followed by the actual setting e.g. **wifi set method dynamic**. Settings are always changed in RAM. Command **s d** reports all current settings in RAM. To make the settings permanent, issue command **s w** (short for **settings write**).

Note that the **settings** command group doesn't change actual settings. These commands manage the settings as a whole.

The commands are documented and implemented in source file **cmd_proc.ino**.

Here's a list of recurring parameters. Parameters that are specific to a command are explained with the command.

```
Recurring parameters:

  IPV4AD  IPv4 address, for example: 192.168.1.1
  YEAR    1970..2225
  MONTH   1..12
  DAY     1..31
  HOUR    0..23
  MINUTE  0..59
  SECOND  0..59
  YN      yes|y,no|n
```

Here's a list of all commands.

```
close                              Close the command interface (network connection only)
```

```
echocmd|ec YN                      Enable or disable echoing of commands on the command
                                     interface
```

```
sys                                System
∟ reset                            System reset
```

```
wifi|w                             Wifi
∟ start|s                          Start WiFi
∟ stop|p                           Stop WiFi
∟ dump|d                           Dump information
∟ set                              Settings
  ∟ ssid "..."                     Specify the SSID, network name (max. 32 characters)
  ∟ password "..."                 Specify the password (max. 63 characters)
  ∟ local IPV4AD                   Static IPv4 local address
  ∟ subnet IPV4AD                  Static IPv4 subnet mask
  ∟ gateway IPV4AD                 Static IPv4 gateway address
  ∟ dns NUMBER IPV4AD              Static IPv4 DNS address (NUMBER=0..1)
  ∟ method|m                       Method of assigning IPv4 address
    ∟ dynamic|dyn                  Use DHCP
    ∟ static|st                    Use static address
  ∟ hostname "..."                 Specify the hostname (max. 63 characters)
serial|ser                         Serial interface
∟ set                              Settings
  ∟ echocmd|ec YN                  Enable or disable echoing of commands on the command
```

	interface

cmdsrv|**cs**	Command server
∟ **start**|**s**	Start the command server
∟ **stop**|**p**	Stop the command server
∟ **dump**|**d**	Dump information
∟ **close**|**c**	
∟ **all**	Close all client connections
∟ INDEX	Close specific client connection (INDEX=0..3)
∟ **set**	Settings
∟ **echocmd**|**ec** YN	Enable or disable echoing of commands on the command interface

ntp	NTP
∟ **start**|**s**	Start NTP request
∟ **set**	Settings
∟ **method**|**m**	Method of determining IPv4 address of NTP server
∟ **hostname**|**h**	Use hostname for looking up IPv4 address
∟ **addr**|**a**	Use IPv4 address
∟ **hostname "..."**	Hostname of NTP server
∟ **addr** IPV4AD	IPv4 address of NTP server

clock|**c**	Clock
∟ **write**|**w** YEAR MONTH DAY HOUR MINUTE SECOND	Write date and time
∟ **read**|**r**	Read current date and time
∟ **set**	Settings
∟ **adjust** [N] **+**|**-** HOUR MINUTE SECOND	Adjustment applied on time received from NTP server (multiple adjustments, N=1..4, default is 1)
∟ **selectadjust**|**sa** N	Select adjustment (N=1..4)
∟ **dateformat**|**df dmy**|**mdy**|**ymd**	Date format: dmy=day-month-year, mdy=month-day-year, ymd=year-month-day
∟ **hourformat**|**hf 12**|**24**	Hour format: 12=12-hour format, 24=24-hour format
∟ **hourlz**|**hlz** YN	Show hour with leading zero y/n

main|**m**	Main function
∟ **ntprefresh**|**nrf**	Perform an NTP request and refresh clock date and time
∟ **resetalarm**|**ra**	Reset alarm if active
∟ **set**	Settings
∟ **wificonnect**|**wc** YN	Automatically connect to wireless network y/n
∟ **ntprequest**|**nrq** YN	Periodically perform an NTP request y/n
∟ **ntprefresh**|**nrf** HOURS MINUTE SECOND	NTP refresh period (HOURS=0..255)
∟ **cmdsrvstart**|**css** YN	Automatically start command server y/n

∟ **gm** N	GM tube, N=1..2
∟ **set**	Settings
∟ **attached**|**att** YN	The tube is physically attached y/n
∟ **accperiod**|**ap** N	Period for reporting accumulated value (minutes)
∟ **deadtime**|**dt** N	Dead time (us)
∟ **cnvfactor**|**cf** N	Conversion factor
∟ **threshold**|**th** N	Threshold value for alarm (nanoSv/h)

buzzer	Buzzer & LED
∟ **alarm**|**al** YN	Turn on or off the alarm sound
∟ **set**	Settings
∟ **enabled**|**ena** YN	Enable or disable the buzzer

```
∟ thingspeak|thsp                 ThingSpeak client
  ∟ set                           Settings
    ∟ gm N                        GM tube, N=1..2
      ∟ writekey|wk "..."         Write API key
      ∟ fieldname|fn "..."        Field name
```

```
opensensemap|osmap                OpenSenseMap client
∟ set                             Settings
  ∟ gm N                          GM tube, N=1..2
    ∟ senseboxid|sbid "..."       SenseBox ID
    ∟ sensorid|sid "..."          Sensor ID
```

```
settings|s                        Settings
∟ write|w                         Write settings from RAM to EEPROM
∟ read|r                          Read settings from EEPROM to RAM, verify, apply
                                    default settings if invalid
∟ reset                           Reset settings in RAM
∟ clear                           Clear settings in EEPROM
∟ dump|d                          Dump information
```

Time Adjustments

The Environment Monitor receives date and time from an NTP server as Coordinated Universal Time (UTC). The NTP protocol supplies no information about time zones or daylight saving time (DST). To display the time correctly for your physical location, the Environment Monitor software adds or subtracts a time offset to or from the displayed time and date. This is called time adjustment.

The data set for time adjustment consist of hours, minutes, seconds and a flag that indicates addition or subtraction.

The Environment Monitor software maintains four time adjustment data sets. There are two commands for managing time adjustment, **clock set adjust** and **clock set selectadjust**. The former sets up a time adjustment data set, the latter selects the active data set.

To adjust for a timezone, one data set is required. Daylight saving time requires an additional data set. For example, when your timezone is UTC+1, you can adjust the displayed time with these commands:
clock set adjust 1 + 1 0 0
clock set selectadjust 1

To support daylight saving time in this timezone, use a second data set. For example, if your location is timezone UTC+1 and it's the winter months, issues these commands:
clock set adjust 1 + 1 0 0
clock set adjust 2 + 2 0 0
clock set selectadjust 1

You can change the timezone data set in the LCD menu for date and time. Changes will be save to EEPROM.

Buying and Testing Russian Geiger-Müller Tubes

One of the most common types of Russian Geiger-Müller tubes is the quintessential SBM-20 tube which is offered by many eBay sellers. Unfortunately, when ordering SBM-20 tubes on Ebay, you may actually get the older STS-5 or CTC-5 tubes or even damaged or used tubes.

A real SBM-20 tube has usually a yellowish/gold color. Also avoid the SBM-20U which has a connection rod at the anode terminal instead of an end cap. This tube will work with the environmental monitor but is harder to mount as it is not recommended to solder to the anode rod. These are often sold as SBM-20 tubes, and unscrupulous eBay sellers will sell the SBM-20/SBM-20U interchangeably. We had a positive experience with eBay seller any-devices (https://www.benl.ebay.be/usr/any-devices).

In order to test an SBM-20 tube you can do the following:

- Visually inspect the tube for mechanical damage (dents, loose end caps, ...).
- Shake the tube gently. A rattling noise may indicate a defective tube.
- Measure with a multimeter the resistance between the anode and the cathode of the tube. You should not measure any resistance (infinite resistance) between both terminals. Also check that there is an electrical connection between the cathode and the metal wall of the tube.
- Now put the tube into the environmental monitor and turn on the buzzer and the LED. You should hear shorts beeps now and then. The LED should also flash in the same rhythm.
- If you have a radioactive source, you can hold it in the vicinity of the Geiger-Müller tube. The rate of the beeps should increase noticeably.
- Measure the background radiation again and slowly heat the Geiger-Müller tube using a hair dryer to approximately 40 to 50°C. The number of beeps per minute (CPM, counts per minute) should not increase.

The SI-1G and the SI-3BG are glass Geiger-Müller tubes. The SI-1G has more or less the same sensitivity than the SBM-20 but is sensitive to light. The SI-3BG is a smaller and very insensitive Geiger-Müller tube. It is only usable for the detection of high radiation levels that would saturate a more sensitive Geiger-Müller tube. We didn't test the environmental monitor with these tubes and do not recommend them, except for special applications like a high radiation alarm.

If you prefer using new or non-Russian Geiger-Müller tubes you can also contact LND in the United States. They offer a range of high quality tubes but the prices are unfortunately quite high starting at around $100. We tested the environmental monitor with the LND 71217 and 78017 tubes with good results. The LND 712 is an option if you want to detect alpha particles and weak beta radiation too. It is less suitable for environmental monitoring.

Radioactive check sources

In order to test the environmental monitor, a radioactive check source may be handy. The best check sources available are exempt quantity radioactive disc sources containing specific isotopes. A distributor of these sources is Spectrum Techniques (https://www.spectrumtechniques.com).

Unfortunately, these are difficult to get especially if you live in a country that requires you to apply for an import license for such radioactive sources.

Another possibility is uranium glass which is still produced in the Czech Republic. This is also safe to handle as the uranium salts are contained by the glass. Look for opaque yellowish-green colored glass beads as these are the most radioactive. We have a positive experience with eBay user hentell001 who sells these glass beads (https://www.benl.ebay.be/usr/hentell001).

Yet another option is the slightly radioactive Quantum Pendant, which is also commonly available on eBay. Look for the round dark grey ones. Please note that it is not as radioactive as the uranium glass but still easily detectable.

Avoid ionization smoke detector sources as these may be illegal to possess and mainly emit alpha particles which cannot be detected by metal wall Geiger-Müller tubes. Also avoid uranium and thorium minerals unless you know what you are doing. These may give off radioactive dust if not packed properly and large samples may emanate dangerous quantities of radon gas.

Calibration

In order to calibrate the Environmental Monitor, the 'conversion factor' for the Geiger-Müller tube has to be determined. Mostly, the ‘conversion factor’ of a given Geiger-Müller tube is not exactly known. Unfortunately, a radioactive source with a relatively high and precisely known activity is required to determine this factor. Such sources are not only very expensive, they usually have to be licensed by a nuclear regulation authority...

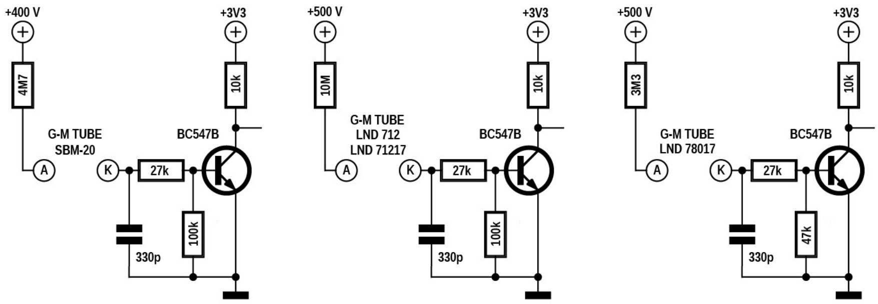
As an alternative, you can trawl the Internet for official radiation monitors in your vicinity and fine-tune the conversion factor of your system to get measurements in line with the official values. For the SBM-20 tube used in our prototype, we obtained good results with a conversion factor of 0.003931.

Internet Connectivity

For environmental monitoring it is best to install the device outdoors in a weatherproof enclosure or cabinet. Make sure the ESP32 DevKitC module has a stable connection to a WiFi router or hotspot. If necessary, use a WiFi extender to provide the Environmental Monitor with a reliable WiFi connection. If the signal quality is poor, the ESP32 DevKitC can loose its connection and a reboot might be necessary to restore WiFi connectivity.

For remote locations where no WiFi signal is available, you can consider using a MiFi router with a prepaid SIM card for internet connectivity. Due to the large power supply voltage range of 9 to 20 V, you can easily power the Environmental Monitor using a solar panel and a rechargeable battery.

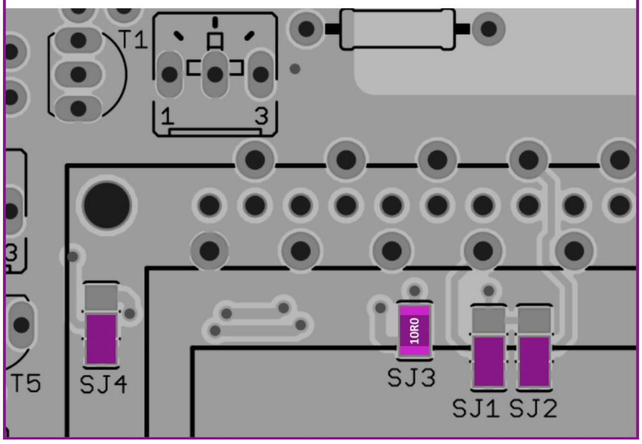
Appendix A – Configuration of the Pulse Detector Circuit for Different Types of Geiger-Müller Tubes



Appendix B – Jumper Settings for different LCD types

Jumper settings for Midas MC21605C6W-SPTLYI-V2 LCD Display

SJ1: bottom position
SJ2: bottom position
SJ3: mount a 10 ohm SMD resistor 0805 size
SJ4: bottom position



Jumper settings for RayStar RC1602B5-LLH-JWV LCD Display

SJ1: bottom position
SJ2: bottom position
SJ3: open
SJ4: top position