Networking Enhancement for a GM1358 Sound Level Meter (the swiss army knife)

Digest

There is a large choice of sound pressure level meters on the market, from pretty cheap to awfully expensive ones. The cheap ones have frequently a sufficient accuracy for many purposes (albeit not being suitable for a legal enforcement). Most of them have however either no, or extremely primitive reporting abilities.

The purpose of this development is to provide networking and reporting abilities as close as possible to IEC 61672-1:2013 and DIN 45643 residential Aircraft noise (cf §Noise Metrics), to an extremely cheap sound pressure level meter GM1358, by adding a ESP8266 WiFi microcontroller to it.

In the first variant sound pressure level meter + WiFi adapter + Online Dashboard the total value of the bill of material will be below 30€ !

The ESP8266 microcontroller will be small enough to fit into the original case of the GM1358 and the requested soldering will be limited to three wires.
With soldering skills, you can do the job in less than 10 minutes.

Your modified GM1358 will then provide USB and WiFi connectivity and be programmable to do the coolest things that only high-end devices will provide:

Evaluation of the noise level according to following time response standards (simultaneously):

- Fast (Attack t=125mS, Decay t=125mS)
- Slow (Attack t=1S, Decay 4,3dB /sec)
- Impulse (Attack t=125mS, Decay 2,9dB /sec)
- Real peak value by the minute (125mS resolution, not the maximum of readings)
- Background level (t=2000s, excluding NAT)

Statistics according to residential aircraft noise standards:

(steady noise equivalents)

- Leq 1 minute
- Leq for each hour of the day
- Leq for 24h
- Leq daytime 06:00 to 22:00
- Leq nighttime 22:00 to 6:00
- Leq 22:00 to 24:00
- Lden

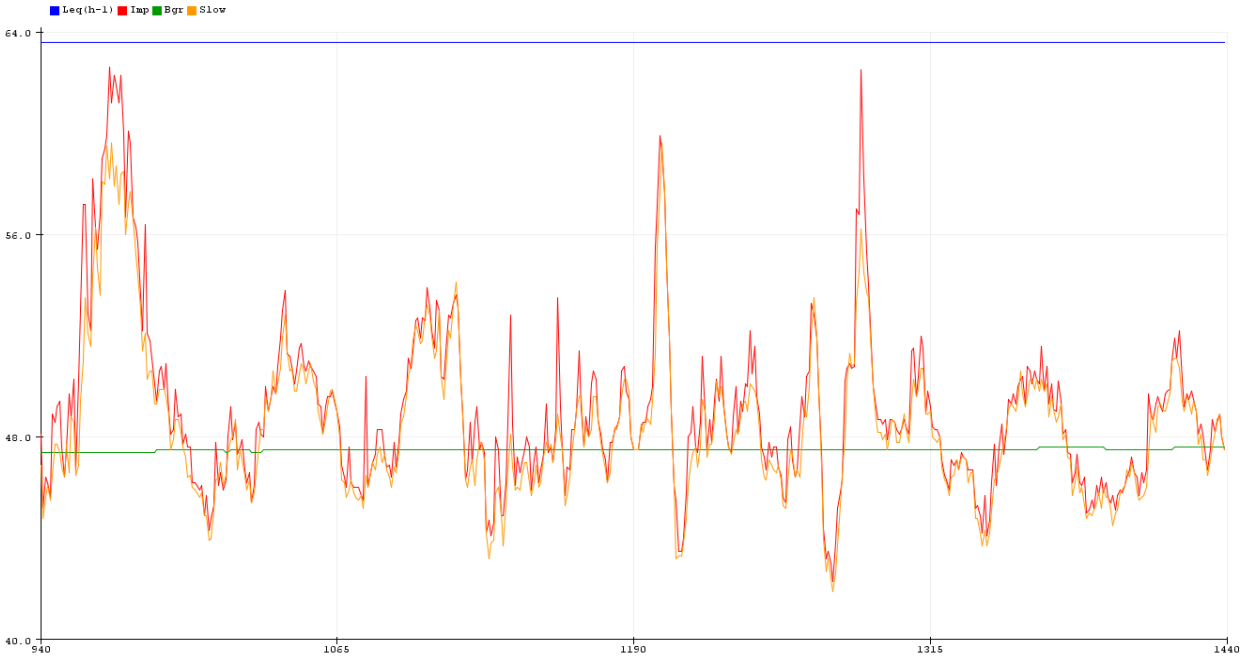
(number above threshold)

- NAT for each hour of the day
- NAT for 24h
- NAT daytime 06:00 to 22:00
- NAT nighttime 22:00 to 6:00
- NAT 22:00 to 24:00

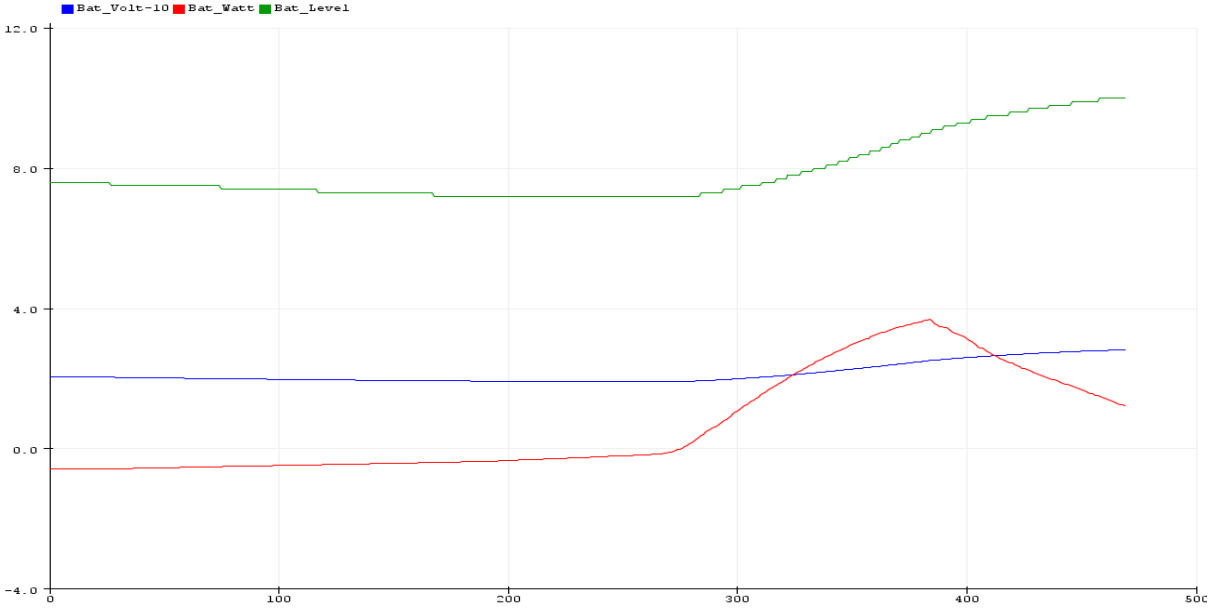
Additionally the program can grab weather information from openweathermap.org and provide the corresponding meteorological conditions.

a) You can report all this information over the USB port using a terminal program or the Serial Monitor of the Arduino IDE.
Over the Serial Plotter you can get a graphical output of the noise or battery evolution history:

USB Sound Plotter from the Arduino IDE:



USB Battery Plotter from the Arduino IDE:

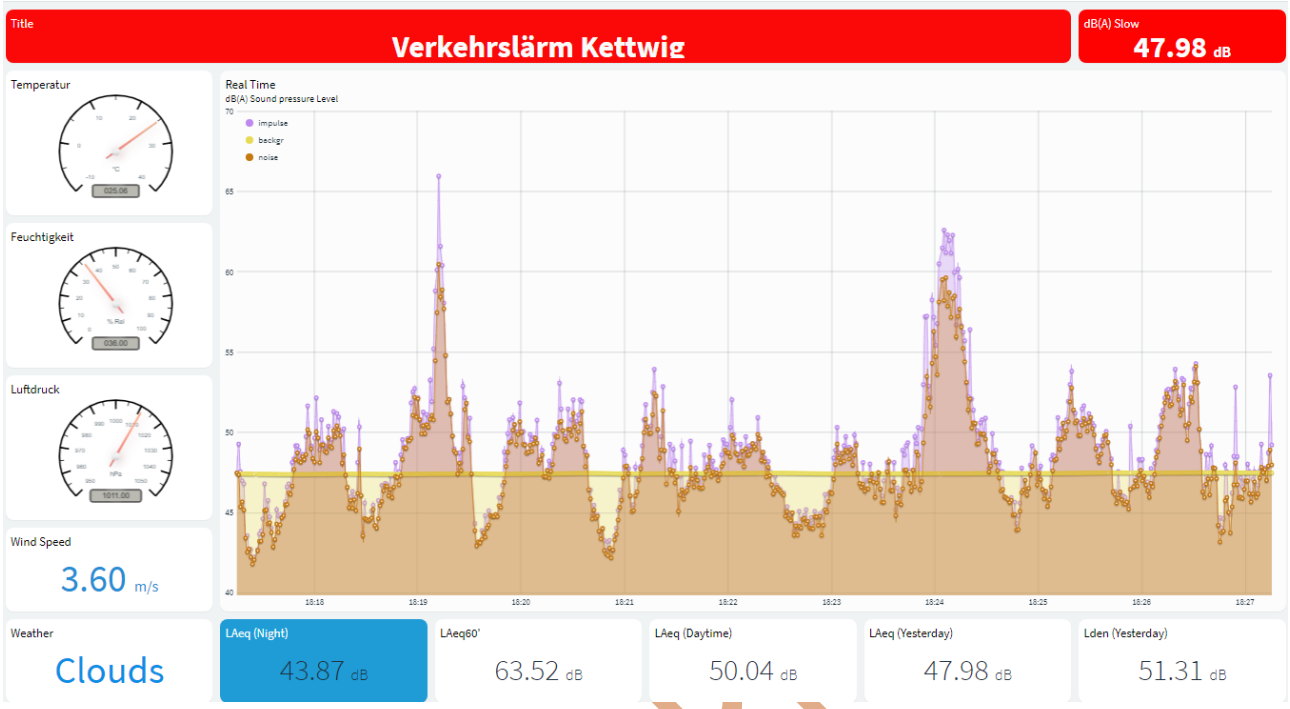


The plotter and reporting abilities of the Arduino IDE are however limited and you can only have one output at a time.
With a free cloud service as e.g Thinger.io, much more features can be used.

Cloud service Thinger

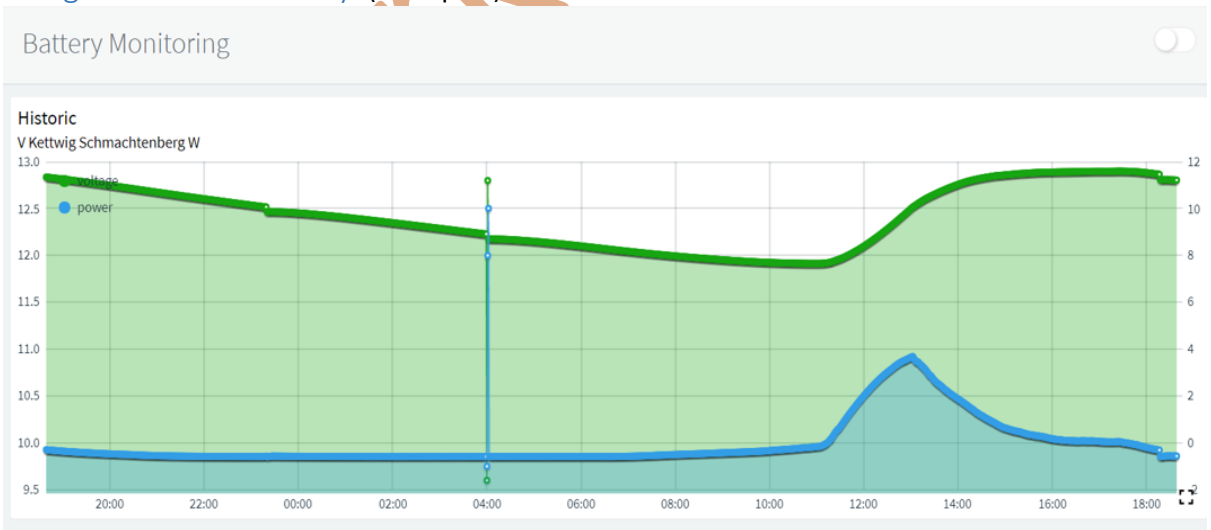
You can register free to the Cloud service Thinger.io to plot information in a very versatile way.
You then can get fast real-time dashboards (that build up over time on screen) and send information to data buckets from which you get historical data (immediately available)

Thinger real time sound plotter and weather information (example1)



With solar hardware and a separate casing, you may also build the system with the ability to be solar powered including a solar power monitor to report all information about the battery condition and the power fed by the solar panel.
You will then operate the system without electrical connection to your computer, using the Cloud service Thinger.io.
Full extension with weather report, solar power report, noise level (needs solar hardware):

Thinger historical battery (example2)



You can also build a split system with the SPL meter (+ the solar power circuitry) being located outside gathering and transmitting the sound (and battery) values over WiFi/ UDP
(Long Range LoRa is planned)
to another bare ESP8266 located inside that will provide the statistics over USB/ Thinger.

Thinger summary dashboard (example3)

Summary Dashboard

Hourly NAT (Number Above Threshold, ongoing)

00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h
1	5	0	0	0	0	0	1	6	7	10	6
12h	**13h**	**14h**	**15h**	**16h**	**17h**	**18h**	**19h**	**20h**	**21h**	**22h**	**23h**
12	9	8	5	5	14	4	3	4	4	1	0

Hourly Lequ (Steady Noise Level Equivalent, ongoing)

00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h
38.14 dB	47.77 dB	42.38 dB	42.13 dB	38.25 dB	40.43 dB	52.97 dB	50.02 dB	54.31 dB	57.91 dB	58.17 dB	55.71 dB
12h	**13h**	**14h**	**15h**	**16h**	**17h**	**18h**	**19h**	**20h**	**21h**	**22h**	**23h**
60.65 dB	60.39 dB	57.64 dB	53.98 dB	55.24 dB	61.50 dB	57.31 dB	48.51 dB	45.58 dB	47.23 dB	49.61 dB	39.63 dB

Hourly Battery Load Balance (ongoing)

00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h
-0.52 Ah	-0.52 Ah	-0.52 Ah	-0.52 Ah	-0.52 Ah	-0.52 Ah	-0.54 Ah	-0.39 Ah	-0.09 Ah	0.11 Ah	-0.02 Ah	0.20 Ah
12h	**13h**	**14h**	**15h**	**16h**	**17h**	**18h**	**19h**	**20h**	**21h**	**22h**	**23h**
0.90 Ah	1.78 Ah	1.72 Ah	1.51 Ah	1.58 Ah	0.54 Ah	0.00 Ah	-0.35 Ah	-0.52 Ah	-0.52 Ah	-0.52 Ah	-0.52 Ah

Midnight Rec: Lequ , Lden (Level DayEveningNight) , NAT , Battery

Leq Day (24h)	Leq Daytime	Leq Nighttime	Leq 22-24h	Lden	NAT (24h)	NAT Daytime	NAT Nightime	NAT 22-24h	Volt Delta@4h	Bat Today	Bat Yesterdy
49.36 dB	54.82 dB	38.44 dB	44.62 dB	52.69 dB	102	101	1	1	0.12 v	-2.59 Ah	2.24 Ah

Last but not least, and back to the historical roots of the whole concept, the system is able, additionally to Thinger, to transmit over USB a single byte per second according to a proprietary "AK-Modulbus protocol" to a feeder program running e.g on a Raspberry Pi forwarding hourly reports to the European aircraft noise network . http://www.eans.net/EANSindex.php

This network is providing a very long time lobby-independent storage of aircraft/railway noise information managed by residentials, currently totaling about 700 privately and communal operated noise stations throughout Europa.

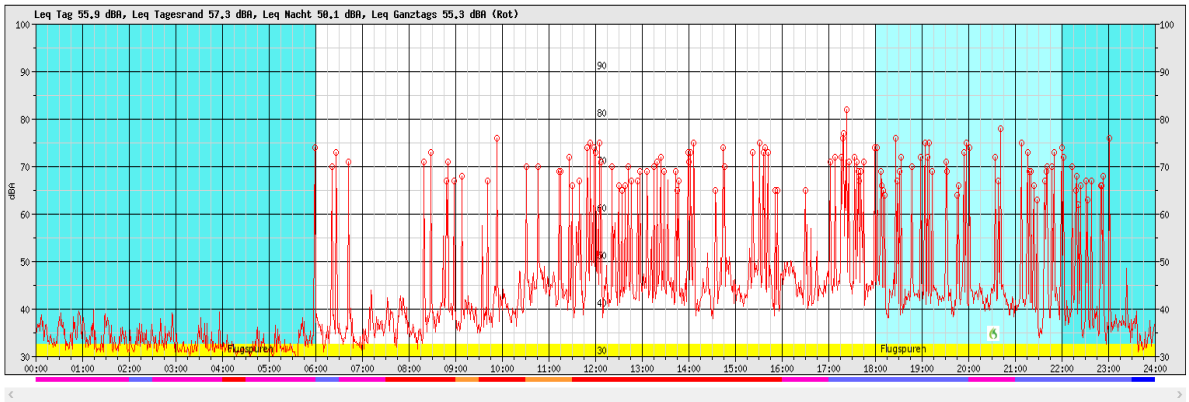
Noise record from DFLD/EANS. (Example4)

Kettwig/Schmachtenberg

02.08.2020 (Sonntag)

Regions Menü

« 02.08.2020 » ⊖ ⊕ Andere Messstation

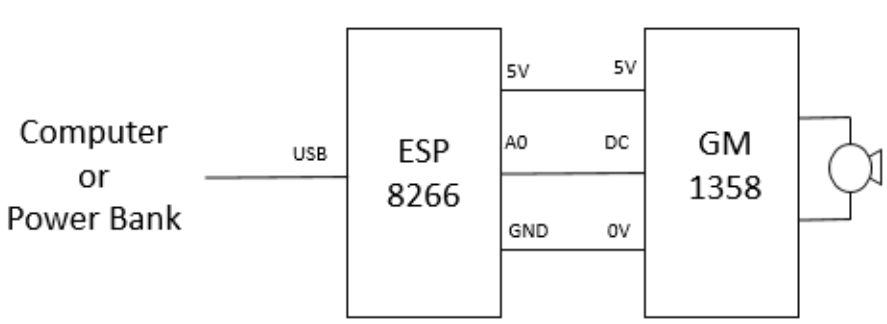


Energy considerations

The built-in 9V alkaline battery block, provides about 24 hours of operation (@ 11mA).
With the retrofitted ESP8266 the total consumption will be about 48 mA, which will drain the battery block within about 8 hours only.
Fortunately, you can also power the system over the USB socket of the ESP8266 and provide over two days of continuous operation from a 3000mA lithium 18360 battery or considerably more from a 12V/12Ah Lead-Acid block.

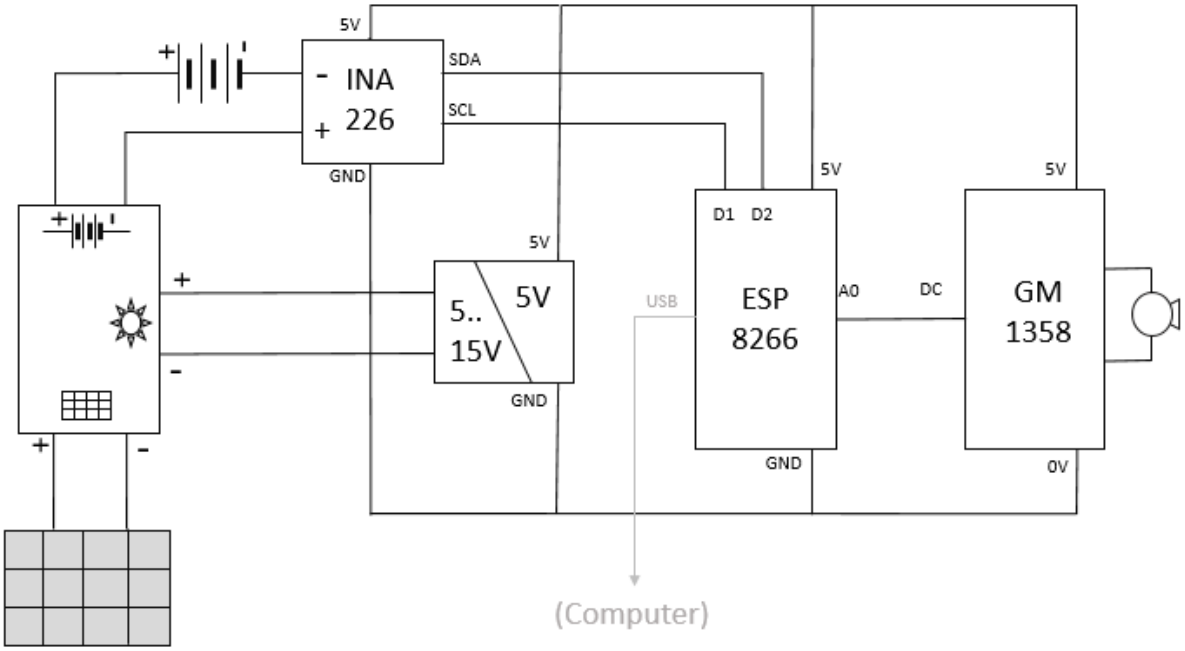
Hardware schematics

Simple version



Solar Powered Version with battery reporting

If you want a 24/24/365 off-grid operation on solar power, depending on your location (e.g. at a 45° latitude) you will need a 12V 30Ah battery and (at least) a 20W solar panel, you can hardly imagine how little energy such as solar panel still delivers on a rainy winter day!



Bill of material

Simple version

You will need exactly that model of Sound level meter: GM 1358
Do not order another model!
That one has the unique feature of providing a linear 0..1V DC signal output that is tied to GND
and the ability to autostart when powered with 5V.
It can interface with an ESP 8266 with only 3 wires.
You can find some devices on eBay.com, but you will find the best offers on AliExpress to largely varying prices.
Anything below 30€ is OK…

RZ GM1358 30-130dB Digital sound level meter meters A/C FAST/SLOW dB screen New

★★★★★ 5.0 ⌄ 2 Reviews 6 orders

€ 16,80 ~~€ 19,76~~ -15%

Ships From:

CHINA | Russian Federation

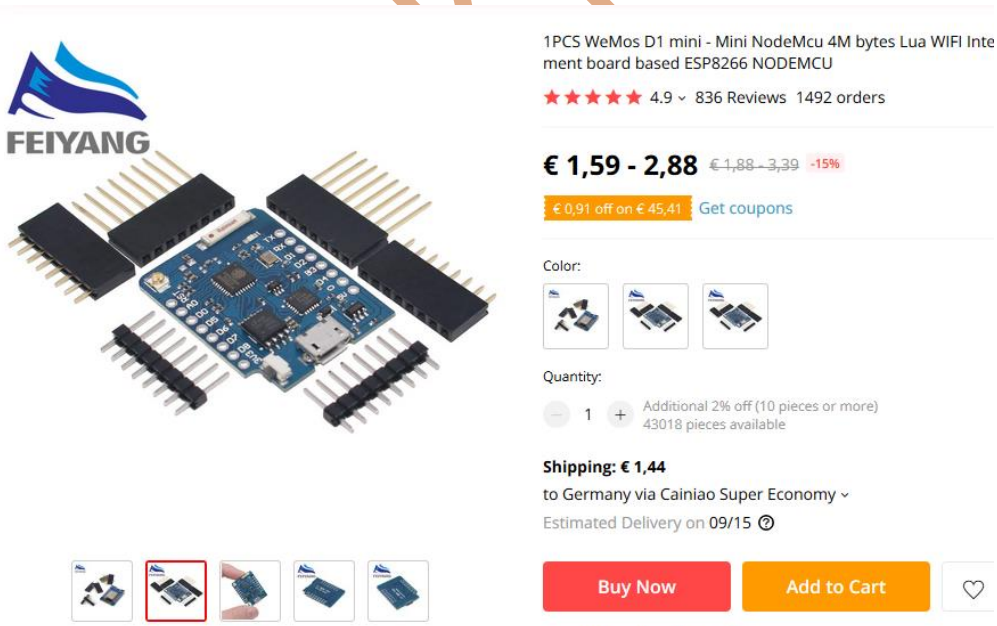
Quantity: 1 +

Additional 3% off (3 pieces or more)
484 pieces available

Please select the country you want to ship from

Buy Now | Add to Cart

https://www.aliexpress.com/item/1647511133.html



1PCS WeMos D1 mini - Mini NodeMcu 4M bytes Lua WIFI Internet board based ESP8266 NODEMCU

★★★★★ 4.9 ⌄ 836 Reviews 1492 orders

€ 1,59 - 2,88 ~~€ 1,88 - 3,39~~ -15%

€ 0,91 off on € 45,41 Get coupons

Color:

Quantity: 1 +

Additional 2% off (10 pieces or more)
43018 pieces available

Shipping: € 1,44

to Germany via Cainiao Super Economy ⌄

Estimated Delivery on 09/15

Buy Now | Add to Cart

90-Day Buyer Protection
Money back guarantee

https://www.aliexpress.com/item/32831353752.html

Option Solar power & monitoring

Free shipping INA226 IIC interface Bi-directional current/Power odule 226 0.01Ohm 0.1Ohm

★★★★★ 4.9 ⌄ 53 Reviews 110 orders

€ 1,30 ~~€ 1,45~~ -10%

Instant discount: € 0,91 off per € 45,41 ⌄

€ 0,91 off on € 91,72 Get coupons

Color:

Quantity:

1 + Additional 1% off (2 pieces or more)
19070 pieces available

Shipping: € 1,15
to Germany via Cainiao Super Economy ⌄
Estimated Delivery on 09/15

Buy Now | Add to Cart | ♡ 36

90-Day Buyer Protection
Money back guarantee

https://www.aliexpress.com/item/32830512534.html

Professional Dual USB 12V/24V 10A Solar Panel Controller Battery Charge Regulator LCD Display Regulator Solar Controller

★★★★★ 4.9 ⌄ 14 Reviews

BRANDS SHOPPING WEEK Ends in 2 d 19 : 15 : 03

€ 1,98 ~~€ 2,86~~ -31%

Instant discount: € 0,87 off per € 25,10 ⌄

+ € 6,93 off per € 129,78 | € 0,87 off on € 18,17 Get coupons

Ships From: China

China | Poland | United States | Spain | Australia | Russian Federation | France | Italy

Color: Dark Blue

Quantity:

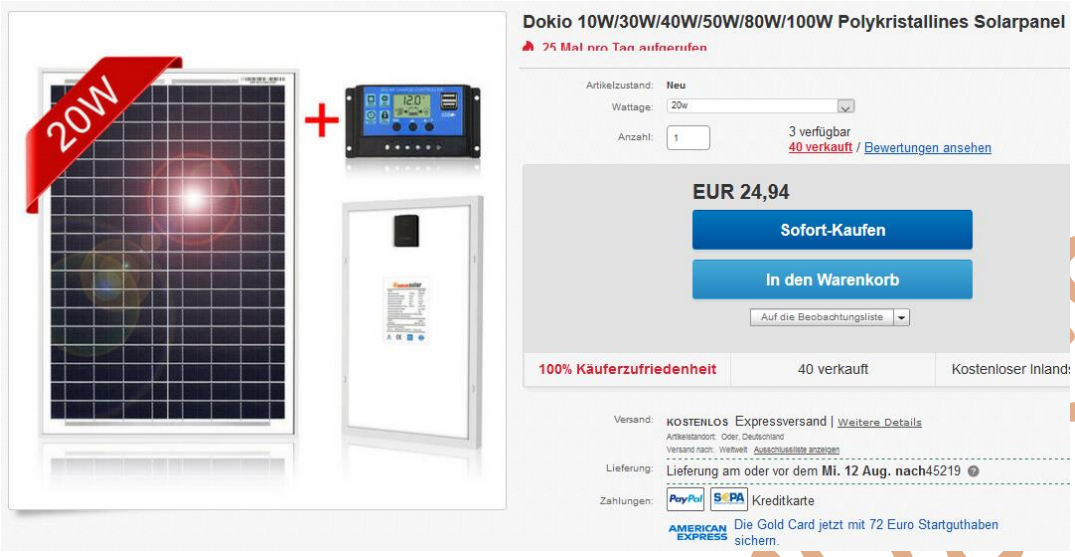
1 + 768 pieces available

Shipping: € 2,17

https://www.aliexpress.com/item/4000715058222.html
for ~4,20€ incl shipment or from Germany:
https://www.ebay.de/itm/Solar-Laderegler-Regler-Solar-Panel-Controller-Regulator-Mit-USB-PWM-12V-24V/402367023023
for ~8€ incl. much faster shipment

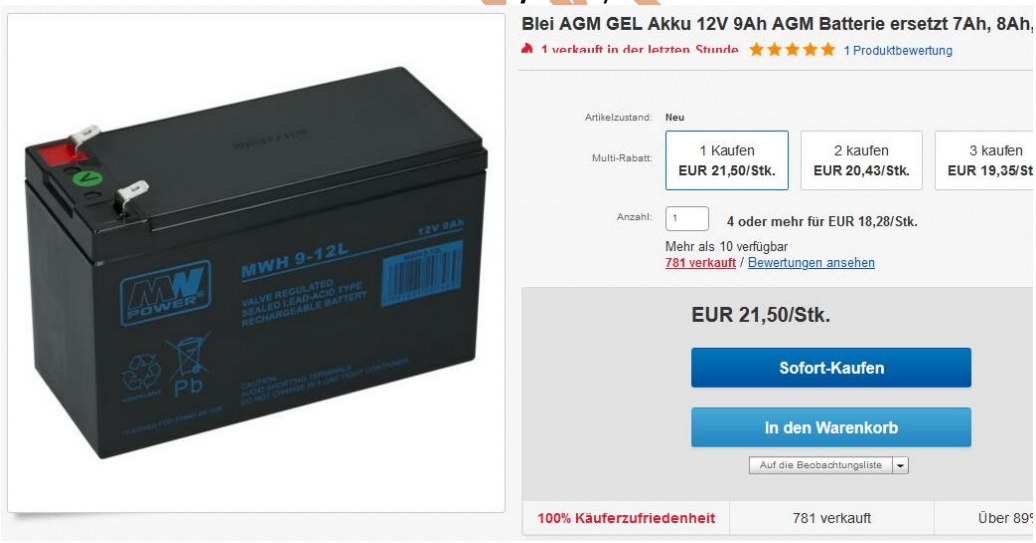
Order a 12v (17v peak) **20W or more solar panel** locally.
Take care to order solar suited batteries and solar panels with a **glass front and an aluminum frame**.
Avoid cheap solar stuff with resin front; they will decay within a few months in bright sun.

Here are some examples from Germany:



(this combo-offer already includes the solar controller and even the USB 5V power supply)
This 100W panel is however much better:
https://www.ebay.de/itm/Solarpanel-Solarmodul-100Watt-150-Watt-12V-12Volt-Solarzelle-Mono-Monokristallin
currently for 53€ incl. shipment.

Order a **12V 12Ah or more lead acid battery** locally.

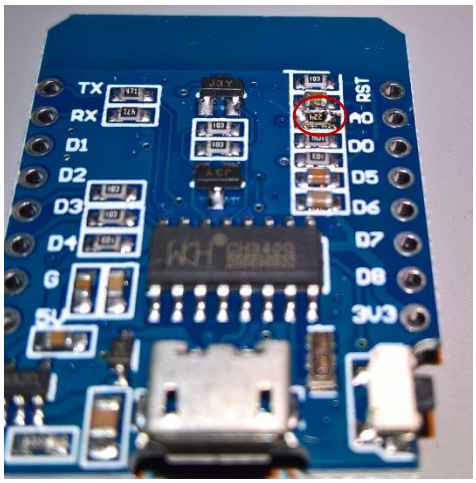


You may use an existing car battery if you have one in acceptable condition, but do not buy one:
car batteries are not optimized for longer periods of operation with a voltage below 13,8v and will decay rapidly.

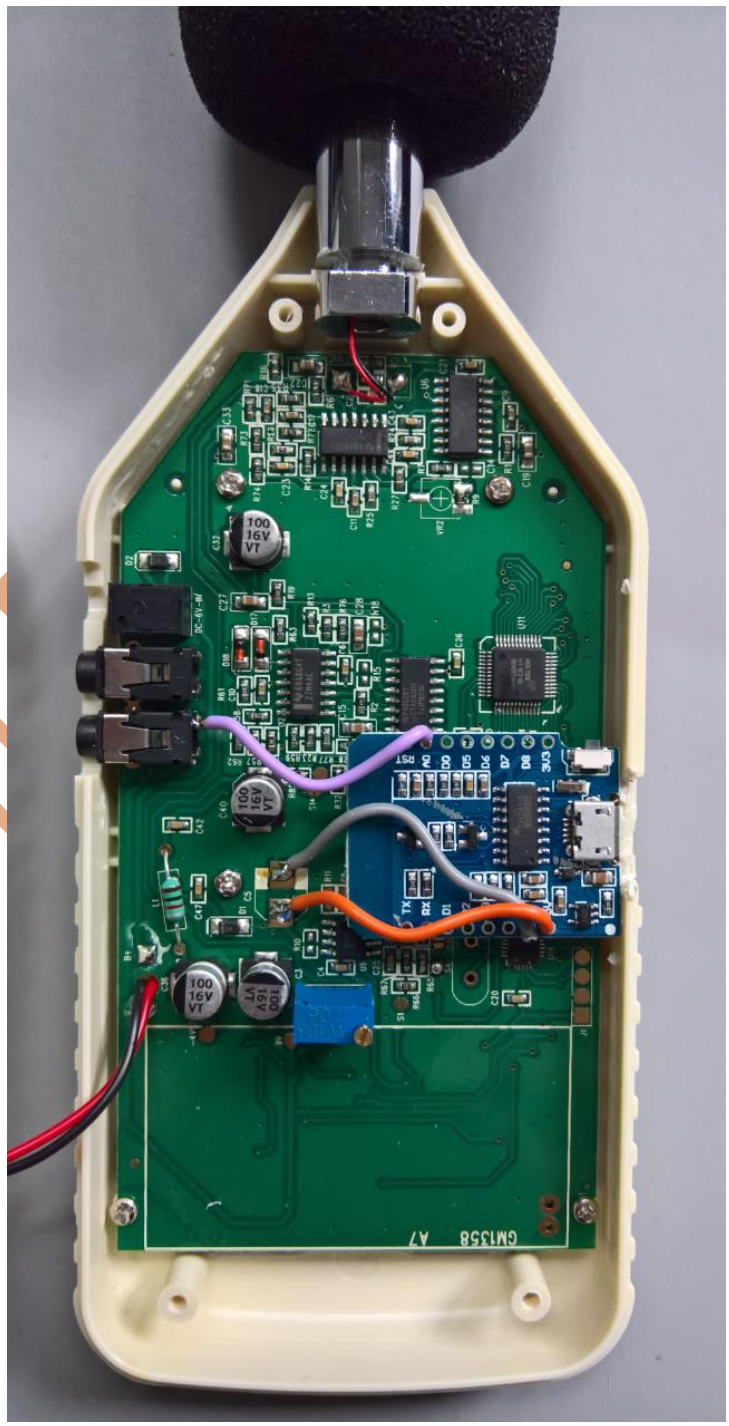
Hardware #1: Modification of the GM1358

Simple Version

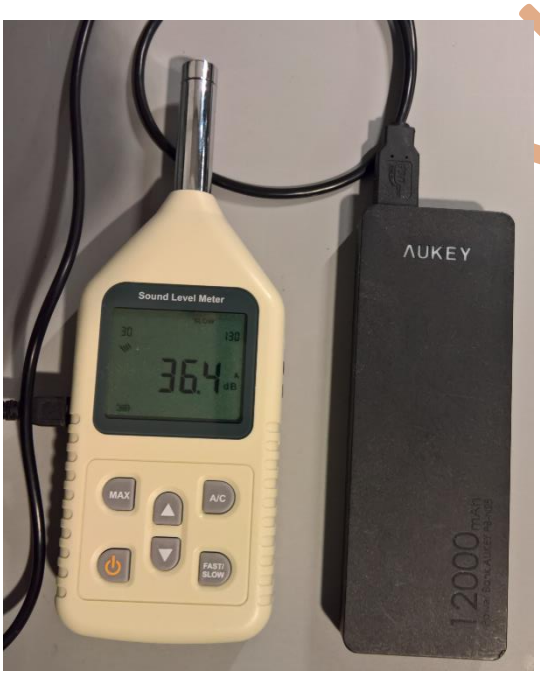
The GM1358 is easy to dismantle, remove 4 screws: 2 at the top and 2 in the battery compartment and you can open the casing.
The first thing might sound scaring but it is easier as it looks: we need to bridge one very tiny resistor on the ESP8266 to change the input range from 0..3,3V to 0..1V.



Take a single strand of electrical wire and solder it to bridge the resistor close to A0 as shown:



Solder the grey wire to GND and the pad on the PCB board, solder the orange wire to GND and the pad on the PCB board, solder the pink wire to A0 and the inside contact of the DC jack
Grind the plastic case to free room for the USB plug.
Glue the ESP8266 as shown.
Screw the case back. Done!



Hardware #2: Replacement for AK-Modulbus Converter

http://arduino.esp8266.com/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html#arduino-ide

The software can be used in conjunction with an AK-Modulbus Sound-Transducer instead of the GM1358.
https://www.ak-modul-bus.de/stat/messmodul_laerm.html

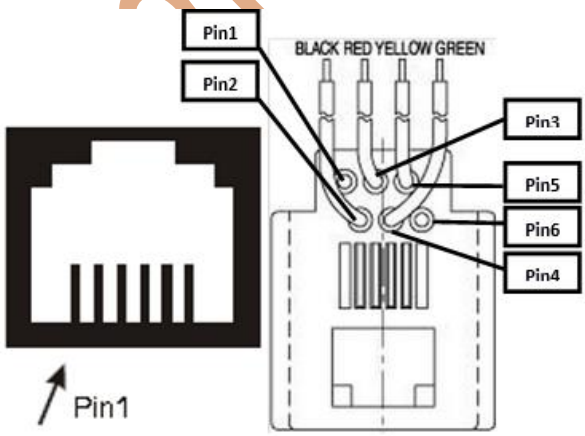
It will then replace and emulate (dramatically improve) the dB-Online interface.

You will need a RJ12 prolongation and cut the male connector.
If you use a regular German standard cable (usually with reversed wiring), you will wire the:

- red(brown) wire (pin3) to A0
- green wire (pin4) to ground
- yellow wire (pin5) to +5V

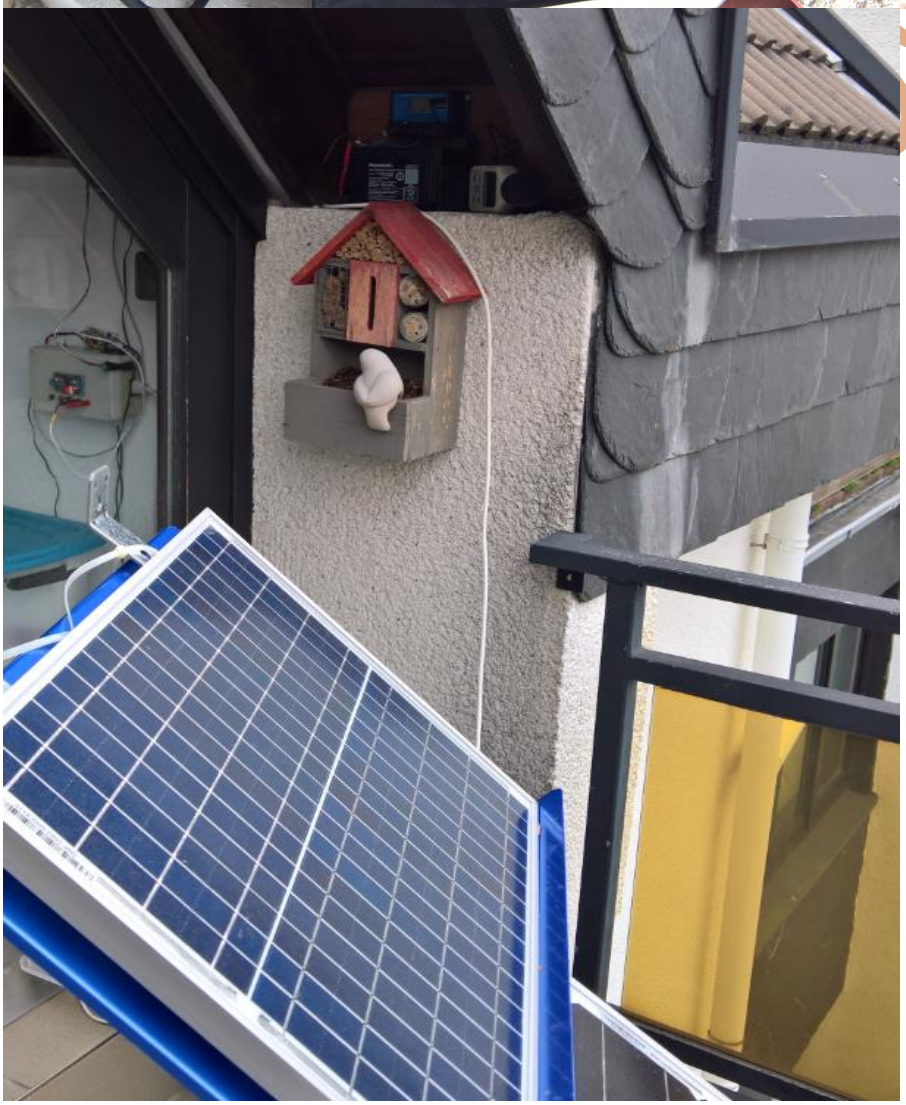
and discard the other wires.
If you appear to have the (unusual) direct wiring, you will wire the:

- green wire (pin3) to A0
- black wire (pin4) to ground
- red wire to (pin5) +5V



You will then save the dB-Online interface, the power supply and the serial-to USB converter.
… and a lot of room and cable mess.

Hardware #3: Simple Solar powered GM1358 under roof



Hardware #4: Full Outdoor / Solar GM1358

to be described in detail later, I am changing that design…

Software

To operate the noise station you will need a WLAN connection and internet connectivity.
You will need to upload the program to the ESP 8266.
For that, if you do not have it already, you should install the Arduino programming Interface (IDE) from www.arduino.cc <http://www.arduino.cc>, on your favourite PC, and then install the ESP framework.
Cf these instructions:
https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/
and
http://arduino.esp8266.com/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html#arduino-ide

Optionally you could register to two free services:
The first step will be to create a free account at www.thinger.io <http://www.thinger.io>
(you can manage two devices free of charge) and
optionally create a free account at www.openweathermap.org <http://www.openweathermap.org> if you want to get weather information for your location.

The current files of my program are hosted on GitHub:
https://github.com/rin67630/Swiss-Army-Knife-for-GM1380-Sound-Pressure-Meter
You may download all .ino files in a folder of your choice and start the Arduino IDE:

I have built the program on the top of my framework ESP-Krarajan on GitHub: https://github.com/rin67630/ESP-Karajan , which takes care of ancillary tasks like, booting to the network and providing scheduling and timing functionalities.

The Arduino IDE provides tabs to split the program into well-structured subparts, so you can jump easily during development between every subpart:

a0) parameters and options to use the program *(the only part you really need to modify)*
a1) libraries and global variables used
b) different functions used in the program
c) setup process
d) menu
e) data processing
f) display (not used here...)
g) serial reports
h) wireless processes
k) finally the scheduler itself which will periodically start the routines listed from d) to h).

Additionally two tabs with comments only are added for convenience:
x_ReadMe and y_ParkedCode, where you can put reminders and code examples.

In my programming style, I deliberately refrain to use too abstract c++ concepts in order to make the program accessible to the majority of people with a minimum of Arduino experience.
I also did put a great attention to comment my code indicating the reason why I have done most steps in that way.

Basically, unless you know exactly what you are doing, you only need to make changes to the Tab "a0 parameters"
The software is written to operate on different hardware configurations and can gather data from different sources.

```
#define HOST_NAME "GITHUB"

// ***Functional Configuration***
#define WEATHER_SOURCE_URL   //_URL _NONE             Change end accordingly
#define BATTERY_SOURCE_UDP   //_INA _UDP _NONE        Change end accordingly
#define SOUND_SOURCE_ANAIN   //_ANAIN _URL _UDP _NONE  Change end accordingly

#define THINGER            //(Comment out, if no thinger used)
#define WRITE_BUCKETS      //(Comment out, if this is the second device for Thinger)

#define Console0 Serial      // Port for user inputs
#define Console1 Serial      // Port for user output
#define Console2 Serial1     // Port for midnight report e.g. on thermal printer
#define Console3 Serial      // Port for boot messages
#define Console4 Serial      // Port for AK-Outputs
#define SERIAL_SPEED           9600 //9600  115200 230400

#define PUBLISH_REPORT       // Issue events&midnight reports to UDP Port + 1, comment out else
//#define PUBLISH_DFLD       // Issue DFLD byte to UDP Port, comment out else
//#define PUBLISH_BATTERY    // If this is the battery master, comment out else
//#define PUBLISH_SOUND      // If this is the sound master, comment out else
#define UDP_TARGET "192.168.188.43"  // RasPi 3a Bare
#define UDP_PORT   4210

//  ***Credentials***
#define SMARTCONFIG  // (WiFi Credentials over GogglePlay/Apple App SmartConfig)
// alternatively to Smartconfig App, you can comment out Smartconfig
// and enter your credentials to initalize for a new WiFi
//#define WIFI_SSID          "SSID"
//#define WIFI_PASS          "Password"
#define wifiMaxTries         30
#define wifiRepeatInterval   100

#define OPEN_WEATHER_MAP_APP_ID      "208085abb5a3859d1e32341d6e1f9079"
#define OPEN_WEATHER_MAP_LOCATION_ID "2928810"
#define OPEN_WEATHER_MAP_LANGUAGE    "de"
#define OPEN_WEATHER_MAP_UNITS       "metric"

#define DFLD_REGION  "004"
#define DFLD_STATION "020"

#define THINGER_USERNAME    "User"
#define THINGER_CREDENTIALS "Credential"
#define THINGER_DEVICE      "Device"
```

```
// ***Time zones***
#define NTP_SERVER "de.pool.ntp.org"
#define MYTZ TZ_Europe_Paris
#define TZ   1                            // (utc+) TZ in hours

// ***Acoustical parameters***
#define Ao94 1050   // 747  for AK with offset and 2,5v  1050 for linear 0..1V
#define Ao47 550   //  461  for AK with offset and 2,5v  550 for linear 0..1V

#define  WIND_LIMIT                 10 // upper limit to record NATs
#define  UPPER_LIMIT_DB             78 // upper limit of plots
#define  LOWER_LIMIT_DB             31 // lower limit of plots
#define  EVENT_THRESHOLD_LEVEL      57 // Begin of Exceedance level
#define  MEASUREMENT_THRESHOLD_LEVEL 55 // Begin of measurement level
#define  MIN_EXCEEDANCE_TIME        10 // Minimum duration of an event
#define  MAX_EXCEEDANCE_TIME        60 // Maximum duration of an event
#define  LISTENING_TIME             50 // mimimum time between events

// ***Electrical parameters***
#define SHUNT    40000    // 16666 = 0,1 Ohm +// 0,02Ohm  or 40000
#define AMPERE   5        // 10 or 5
#define SERIAL_SPEED 9600 //9600  115200 230400
#define MIN_VOLT 11.8     // 11.8 for Lead Battery,  9.6 for 3x18360 Lithium
#define MAX_VOLT 14.2     // 14.2 for Lead Battery, 12.8 for 3x18360 Lithium
#define MIN_AMP  -0.8
#define MAX_AMP  +0.8
#define MIN_WATT -1
#define MAX_WATT +20
```

Software Configuration

In the first chapter ***Functional configuration*** you will determine:

a) The host name (for your WiFi and e.g. for Thinger
b) Whether you will be using Weather information, mainly you will use
   _URL to get the information from openweathermap.org.
   _NONE if you don't need the weather information.
c) Configure the source of the battery information:
   _NONE, if you don't have any battery information.
   _INA if you have an INA226 module to measure current and voltage
   _UDP if you have in the same network another ESP8266
   running the same sketch as a master
d) Configure the source of the sound information
   _NONE, if you don't have any battery information
   _ANAIN if the sound level is coming from the analog input A0
   _UDP if you have in the same network another ESP8266
   running the same sketch as a master
   _URL if the sound level is coming from a JSON URL of www.dfld.de
   (then you need to uncomment the next line and provide the URL
   that will provide the sound information)

The line #define THINGER is determining, if you are using the cloud services of thinger.io. If you do not, please uncomment that line.
The line #define WRITE_BUCKETS is determining, if this device is allowed to write the buckets. Only one device should write the buckets per account.

The lines with #define CONSOLEx determine where some parts of the sketch are issued.
Normally to Serial, with additional UARTS ports like Serial1 are possible.
Serial1 can be wired to a thermal printer.

In the next chapter ***Credentials***, you will enter the credential information for different services.

The sketch is using SmartConfig a facility from Espressif that enables to configure your WiFi credentials we do not need to recompile the whole sketch.

You have got three possibilities:

a) If you ever have run your ESP8266 with another sketch containing the credentials successfully, then the ESP has stored how to connect in its nonvolatile memory and you just can't go on without caring for credentials.
b) If you never have run your ESP8266 with credentials, you can comment out #define SMARTCONFIG and enter your credentials in the 2 next lines, after having removing the slashes. After successful connection, it is recommended to return to the initial active SMARTCONFIG and for security remove your credentials from the sketch
c) You can reconfigure your ESP8266 from an Android or Apple smartphone to every local WiFi to which you have access.
   To do that, please download the application ESP SMARTCONFIG from the respective app-store and follow the instructions there.

USB-Serial Menu

The Software provides a simple “single character” command-line menu over the USB serial line.
Commands are given by a single character and executed over [return]
Commands are stackable: you can give several characters then [return]; all commands will be executed in sequence.
Example: U+++ means: Apply 94&47dB Defaults and Increase Offset by 3dB.
Usually an upper case letter sets the function and the lower case reset the function.

Control actions

'Z': //Reset the ESP device
'C': //Apply 94dB Calibration
'c': //Apply 47dB Calibration
'U': //Apply 94&47dB Defaults
'+': //Increase Offset by 1dB
'-': //Reduce Offset by 1dB

Control Display

(abandoned, could be re-implemented, it would need a display, Thinger is more powerful and needs no hardware)

'0': //Display mode 0
…
'3': //Display mode 3

Periodical reports over the USB serial line

'A': //serialPage AK
(this is not a printable report, it issues one byte every second to feed the DFLD website)

'P': //Periodical Reports on 'p': //Periodical Reports off
Options for periodical reports:

'D': //Day Report	'd': //no Day Report
'H': //Hour Report	'h': //no Hour Report
'M': //Minute Report*	'm': //no Minute Report (Battery)
'S': //Second Report*	's': //no Second Report (Noise)
'E': //Event Report	'e': //no Event Report

* these reports are designed to produce serial Plotter compatible results.

Example1 : PDHmsE means: Print Daily, Hourly, no minute, no second, Events
Example 2: p means: stop printing reports.
Example 3: P means: resume printing reports with last options
Example 4: Sd means: now with Second reports without Daily reports.

One shot reports

n.b. these reports stop periodical reports, resume with “P” to return to periodical printing.

'L': // Leq Report by 24h
'N': // NAT Report by 24h
'B': // Battery Report by 24H
'b': // Battery Report (Actual measurements)
‘W’: // Weather report

'?': //List parameters
'~': //List WLAN / Radio settings.

Report examples

Minute Hour and Events

Menu command MHEP:

```
08:55:59.490 -> Bat_Volt-10:1.928 Bat_Watt:-0.593 Bat_Level:7.200
08:56:59.461 -> Bat_Volt-10:1.927 Bat_Watt:-0.593 Bat_Level:7.200
08:57:59.462 -> Bat_Volt-10:1.926 Bat_Watt:-0.593 Bat_Level:7.200
08:58:59.502 -> Bat_Volt-10:1.925 Bat_Watt:-0.593 Bat_Level:7.200
08:59:59.489 -> BatAhBat:0.000 A0dBLEQ:58.8 WindSpeed:0 Direction:1072693248
08:59:59.536 -> Bat_Volt-10:1.924 Bat_Watt:-0.593 Bat_Level:7.200
09:00:59.478 -> Bat_Volt-10:1.923 Bat_Watt:-0.593 Bat_Level:7.200
09:01:59.466 -> Bat_Volt-10:1.922 Bat_Watt:-0.593 Bat_Level:7.200
09:02:47.466 -> PKTm: 09:01:37 PKdB:52.8 ATdB: 50.2 ATsec:48 PK-10dB: 49.7 PK-10sec: 41 NAT:1
09:02:59.489 -> Bat_Volt-10:1.921 Bat_Watt:-0.593 Bat_Level:7.200
09:03:59.488 -> Bat_Volt-10:1.920 Bat_Watt:-0.593 Bat_Level:7.200
```

Menu command L:

```
Leq : for Saturday, 08 August 2020
 Hour   |  00  |  01  |  02  |  03  |  04  |  05  |  06  |  07  |  08  |  09  |  10  |  11  |
 Leq dB | 58.1 | 58.3 | 58.3 | 58.4 | 58.4 | 58.5 | 58.6 | 58.7 | 58.8 | 61.1 | 65.4 | 65.8 |
 Hour   |  12  |  13  |  14  |  15  |  16  |  17  |  18  |  19  |  20  |  21  |  22  |  23  |
 Leq dB | 66.0 | 66.2 | 66.3 | 62.9 | 63.2 | 63.6 | 63.8 | 54.0 | 56.3 | 56.6 | 57.3 | 57.8 |
```

Menu command N:

```
NAT : for Saturday, 08 August 2020
 Hour   |  00  |  01  |  02  |  03  |  04  |  05  |  06  |  07  |  08  |  09  |  10  |  11  |
 NAT    |  00  |  00  |  01  |  00  |  01  |  00  |  00  |  00  |  00  |  01  |  24  |  23  |
 Hour   |  12  |  13  |  14  |  15  |  16  |  17  |  18  |  19  |  20  |  21  |  22  |  23  |
 NAT    |  20  |  21  |  19  |  06  |  12  |  09  |  34  |  12  |  13  |  01  |  01  |  00  |
```

Menu command B:

```
Battery History :
 Hour   |   00   |   01   |   02   |   03   |   04   |   05   |   06   |   07   |   08   |   09   |   10   |   11   |
 Bat Ah | -0.045 | -0.047 | -0.048 | -0.048 | -0.048 | -0.048 | -0.048 | -0.049 | -0.049 | +0.000 | +0.000 | +0.000 |
 Hour   |   12   |   13   |   14   |   15   |   16   |   17   |   18   |   19   |   20   |   21   |   22   |   23   |
 Bat Ah | +0.000 | +0.000 | +0.000 | +0.000 | +0.000 | +0.000 | +0.000 | +0.000 | -0.044 | -0.044 | -0.044 | -0.044 |
```

Automated reporting on Serial1 (e.g. on thermic-printer) or on UDP:

This is an example of the ongoing event + midnight summary report

NAT: (Number of Above Threshold in hour;

PKTime: (Peak Time of the event)

Leq4: (Level equivalent for the time defined by max-10dB to max-10dB on the other side)

t10: (time defined by max-10dB to max-10dB on the other side)

Leq3:: (Level equivalent for the time above threshold)

t AT: (Time above threshold)

```
-------------------------------------------------
NAT|PKTime  |PKdB|Leq4|t10|Leq3|tAT|
 09|18:46:07|72.0|67.6|28 |66.9|66 |
 01|19:03:48|72.9|68.5|27 |67.8|66 |
 02|19:16:37|69.8|65.6|23 |65.2|51 |
 03|19:18:56|65.2|62.3|25 |62.3|50 |
 04|19:21:32|66.5|63.6|33 |63.4|69 |
 05|19:31:20|64.6|60.0|17 |60.1|33 |
 06|19:49:01|62.8|59.2|21 |59.3|41 |
 07|19:55:36|69.9|65.1|24 |64.8|53 |
 08|19:57:44|70.7|67.8|25 |67.1|59 |
 01|20:00:22|64.7|61.6|30 |61.6|60 |
 02|20:12:42|63.9|61.2|32 |61.2|63 |
 03|20:36:49|69.5|66.4|29 |66.2|61 |
 04|20:45:26|64.5|61.3|25 |61.4|49 |
 05|20:58:18|68.5|65.2|22 |64.9|48 |
 01|21:10:13|68.2|65.2|34 |64.9|75 |
 02|21:21:39|70.7|66.8|28 |66.2|67 |
 03|21:33:48|72.2|68.6|30 |68.1|69 |
 04|21:38:24|71.7|67.7|24 |67.2|55 |
 05|21:41:08|74.4|71.2|27 |70.5|64 |
 06|21:43:09|70.7|68.0|31 |67.6|70 |
 07|21:45:01|70.1|66.4|23 |65.8|54 |
 08|21:46:41|72.9|69.1|32 |68.8|70 |
 09|21:48:48|71.7|68.9|30 |68.4|68 |
 01|22:02:02|69.2|66.6|25 |66.2|56 |
 02|22:05:02|72.2|68.4|22 |67.2|60 |
```

```
 03|22:07:24|68.3|65.3|28 |65.2|57 |
 04|22:14:10|74.1|68.6|19 |67.7|49 |
 05|22:19:29|69.1|66.5|27 |66.2|58 |
 06|22:24:02|67.0|64.1|28 |64.0|57 |
 07|22:25:55|69.8|65.6|26 |65.2|58 |
 08|22:28:26|69.0|65.5|26 |65.0|59 |
 09|22:33:42|65.6|62.7|26 |62.7|52 |
 01|23:10:48|57.3|54.0|22 |55.0|31 |
 02|23:16:23|59.9|54.6|23 |55.5|30 |
 03|23:23:34|58.8|54.5|51 |55.3|64 |
 04|23:41:10|56.0|51.2|44 |51.5|46 |

Daily Report for
Sunday, 23 August 2020
Hour| Leq  |  NAT  |  Ah    |
00  | 39.3 |  026  | -0.039 |
01  | 39.5 |  000  | -0.037 |
02  | 37.1 |  000  | -0.036 |
03  | 36.8 |  000  | -0.035 |
04  | 37.0 |  000  | -0.035 |
05  | 43.9 |  001  | -0.035 |
06  | 43.4 |  002  | -0.035 |
07  | 52.5 |  010  | -0.040 |
08  | 47.6 |  004  | -0.038 |
09  | 52.5 |  010  | -0.040 |
10  | 52.4 |  017  | -0.037 |
11  | 52.3 |  008  | -0.018 |
12  | 33.2 |  012  | +0.000 |
13  | 57.9 |  012  | +0.000 |
14  | 55.1 |  013  | +0.132 |
15  | 59.0 |  010  | +0.000 |
16  | 58.1 |  015  | +0.000 |
17  | 59.3 |  013  | +0.000 |
18  | 56.2 |  009  | +0.000 |
19  | 53.3 |  008  | +0.000 |
20  | 49.9 |  005  | +0.000 |
21  | 57.1 |  009  | +0.000 |
22  | 54.8 |  009  | +0.000 |
23  | 48.1 |  004  | +0.000 |
Extra hours cf. Man.
25  | 48.1 |  000  | +0.000 |
26  | 49.0 |  197  | -0.295 |
27  | 52.5 |  157  | -0.295 |
28  | 42.1 |  040  | +0.000 |
29  | 52.4 |  013  | +0.000 |

NAT|PKTime  |PKdB|Leq4|t10|Leq3|tAT|
 01|00:48:56|57.3|52.8|47 |53.5|55 |
 01|05:58:36|76.2|73.3|24 |72.4|59 |
 01|06:15:56|70.9|65.8|26 |65.3|60 |
 02|06:21:05|71.2|66.0|24 |65.5|55 |
 03|06:45:38|68.5|65.7|29 |65.5|61 |
 05|08:14:30|56.5|51.8|32 |52.5|36 |
 06|08:17:54|67.2|63.7|18 |63.2|42 |
 07|08:47:40|74.6|69.5|20 |68.2|58 |
 01|09:05:24|71.5|67.9|25 |67.0|64 |
 02|09:09:36|59.0|54.9|51 |55.6|76 |
 03|09:15:44|72.5|69.6|28 |69.1|64 |
 04|09:25:56|72.5|69.0|27 |68.2|66 |
 05|09:32:01|62.2|58.0|37 |58.0|72 |
 06|09:34:02|58.1|52.9|42 |53.6|49 |
 07|09:35:25|57.5|53.7|43 |54.5|57 |
 08|09:40:17|75.6|72.4|30 |71.1|84 |
 09|09:42:43|62.6|57.4|52 |57.9|86 |
 10|09:44:18|59.9|57.0|20 |57.4|35 |
 11|09:46:54|61.5|55.9|39 |56.6|55 |
 12|09:53:19|69.5|56.0|60 |56.5|95 |
 00|09:59:03|59.9|56.0|22 |56.8|30 |
...
```

Norms and Regulations on Residential Noise

Noise Metrics used.

Equivalent sound level (LEQ)

That metric measures the average acoustic energy over a period of time to take account of the cumulative effect of multiple noise events. This could, for example, provide a measure of the aggregate sound at a location that has airplane flyovers throughout the day. LEQ is defined as the level of continuous sound over a given time period that would deliver the same amount of energy as the actual, varying sound exposure.

The Leq metrics and all metrics used in residential noise computations are not simple averages, but are the result of a complex calculation:
the sound level is first delogaritmed (exponential function) to convert decibel into the linear sound pressure, averaged, and subsequently logaritmed again to return the Leq result:

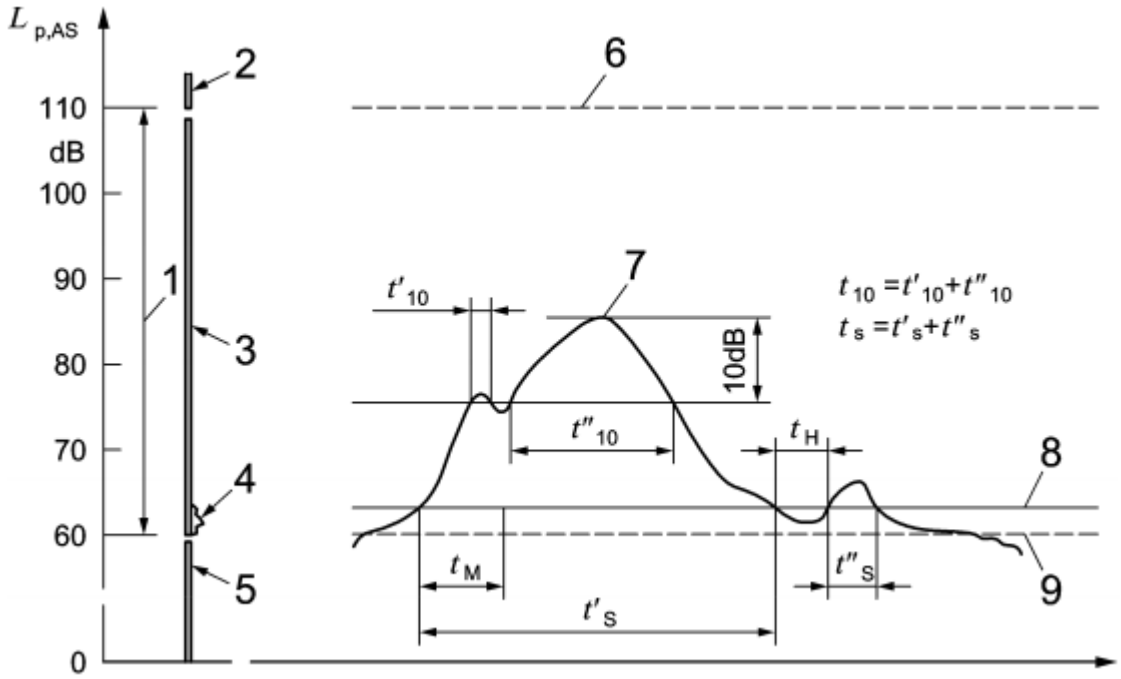
$$L_{p,\mathrm{A,eq},T} = 10\ \mathrm{lg}\left(\frac{t_0}{T}\sum_{i=1}^{N} 10^{L_{p,\mathrm{A,E},i}/10\ \mathrm{dB}}\right)\mathrm{dB}$$

Day-night average sound level (Lden)

That noise metric reflects a person's cumulative exposure to sound over a 24-hour period, expressed as the noise level for the average day of the year based on annual aircraft operations, given an additional 10dB for events occurring during the legal night (from 22:00 to 06:00).
Lden is the standard noise metric used for all FAA studies of aviation noise exposure in airport communities.

Leq4 metric

The noise metric represents (legally unfortunately, not really) all the acoustic energy (a.k.a. sound pressure) of an aircraft noise event delimited by the boundaries of its own maximum level minus 10 dB.
This figure will be the only accepted metric for the noise event, all (really existing and already annoying part below that part are not considered. ☹
The t10 metric is the duration of that event.



Keys used in figure:
1 Primary indication range/dynamic range, 2 Overload range, 3 Range included in the assessment
4 Range not included in the assessment, 5 Range not transferred, 6 Upper limit of the primary indication range/dynamic range, 7 Maximum sound level Lp,AS,max, 8 Measurement threshold level Lp,AS,MSchw
9 Lower limit of the primary indication range/dynamic range
t 10 10 dB-down-time, H Listening time, M Minimum time, t s exceedance time

Leq3 metric

The Leq3 metric represents (unfortunately, not legally) all the acoustic energy (a.k.a. sound pressure) of an aircraft noise event delimited by a given threshold (e.g. 55dB), which is much closer to the real annoyance.
The tAT metric is the duration of that event.

The residual sound

The residual sound metric is the long-term averaged sound level excluding the noise events.
The noise events must (legally) at least exceed the residual sound by 5dB, which is in my opinion ridiculously low. I considered at least 10dB distance from residual sound, to accept the noise event.

Weather impact

The microphone used in the weather station should not put more than 10dB additional noise upon a constant laminar wind strength of 10 m/s.
I have tested my microphones at the Düsseldorf University of technology in a wind funel and the test has proved the design to be compliant.
However according to the regulations, for wind and windgusts above 10m/s, the overflight detection is disabled.
The weather condition is gathered form openweathermaps.org. due to the ost and complexity, I do not yet consider an individual anemometer.

The NAT metric

The number above threshold is the number of detected events according to the Leq4 metric within a given period.
Legally, one considers NAT limits only during night.
Airports usually report NAT for day (6:00 -22:00) and night (22:00 -6:00) only.
I am reporting NAT hourly to identify the most noise impacted hours of the day.

The Noise contours

Noise contours are determined by measuring the very long term equivalent noise level, considering it a period of at least the six-month with the most traffic (usually from May to October). This has the effect of diluting the noise events and reducing the legal noise impact to ridiculously low levels. ☹.

Some helpful items to purchase

To hold on a semi-mobile way a small 20W solar panel:
https://www.thomann.de/de/thomann_orchesterpult.htm ~15€
To hold on a semi-mobile way the electronics and battery:
https://www.thomann.de/de/flyht_pro_uac_universal_alu_case_s.htm ~50€

Table of Contents

(left empty)