GPS logger for speedsurfing

RP6Conrad

6 okt 2022

Overview

An inexpensive GPS logger + display, specifically suitable for speedsurfing. Material costs are less than 50€.

Goals

1. Easy to build yourself with readily available components.
2. Measurements with classic doppler speeds at max 10 Hz (20 Hz with the ublox M9 gps module), suitable for upload to GPS speedsurfing.
3. Completely Open Source, source code is on github : https://github.com/RP6conrad/ESP-GPS-Logger/edit/master/README.md

Specifications

- Approved GPS-logger for upload to GPS-speedsurfing.com
- Logging of the UBX NAV PVT protocol at 1, 2, 5, 10, 15 or 20Hz to micro SD
- Logging in .sbp (locosys format), or new condensed .gpy format.
- Log in .gpx format @ 1 Hz possible.
- Lipo battery 2000 mAh with built-in protection, approx 15 hours autonomy.
- Built-in charge controller, charge via micro USB or wireless.

- Consumption approx. 80 mA, sleep 1.5 mA (version with LEDs), 0.6 mA (without LEDs)
- ESP32 µ-controller 240 MHz, built-in WiFi + bt
- Update software “Over The Air” via WiFi possible.
- All files can be downloaded over the build in webserver. Configuration over the webserver.
- Alternatif download GPS files over WiFi with “ftp” (file transfer protocol).
- E-paper display, easy to read in daylight, update rate 1000 ms.
- Enable/disable with reed switch (magnet) or pushbutton.
- GPS module of your choice, ublox M8N / M9 / M10 engine required.
- Display of current speed, last run, best 2s, 10s, 30 min, 1 hour, avg 5*10s, best 100m, 250m, 500m, 1852m, total distance and alpha 500.
- Configurable km/h or knots

Parts

- Lilygo TTGO T5 ESP32 E-paper development board (15€)
- Micro SD card, max 32 GB, SDHC. SDXC cards will not work !
- GPS module Beitian BN180, BN220 or BN280 (M8), BE180, BE220, BE280 (M10), BK180, BK220...(M9) or other ublox M8N / M9 / M10 based GPS(15€). **The Beitian BN220ZF / BN280ZF is NO GOOD !!** They have a not compatible AT - GPS-chip. The unbranded BN220 chips are not so good as the Beitians, so stay with the branded ones !
- Lipo 2000mAh with built-in protection (12€)
- Wireless inductive charging circuit (3€)
- Reed switch (0.5€) or watersealed Push Button (2€)
- Waterproof box (3€)
- Alternative : GoPro 7/8/9/10 housing (10€)

Lilygo TTGO T5 e-paper

There are several variants in circulation, with minor differences. I know of 3 versions :

1. T5_V2.3.1_2_2.13, with the epaper screen GDEM0213B74, no LEDs, most recent.
2. T5_V2.3.1_2.13, with the epaper GDEH0213B73, or LEDs
3. T5_V2.3.1_2.13, with the epaper GDEH0213BN
4. T5_V2.3.2_2.13, with the epaper GDEH0213BN without serial chip, interface T-U2T needed !
5. T5_V2.4_2.13 with the epaper GDEH0213BN (other PCB, no output 25 !)

These versions have a resolution of 250*122, which is mandatory.

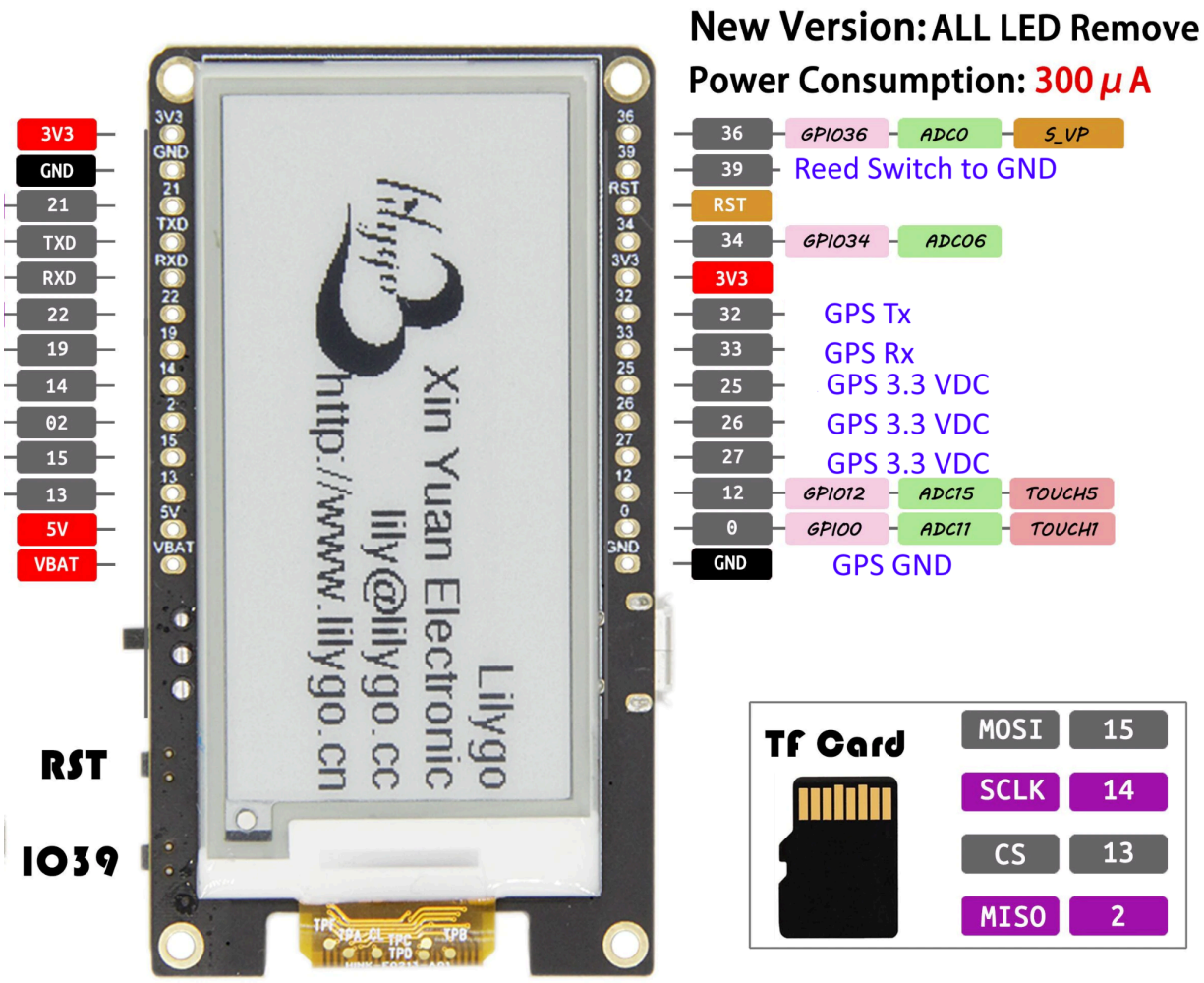
The GDEW0213M21 / GDEW0213T5 has a lower resolution 212*104, and are no good !!!

T5 2.13 inch Product Comparison Chart

SKU	H239	H239-01	H480	H239-02
Grey Level	2	4	4	4
Model	DEPG0213BN Res 250*122	GDEM0213B73 GDEM0213B74 Res 250*122	GDEW0213T5 Res 212*104	GDEW0213M21 Res 212*104
Work Temp.	0℃~50℃	0℃~50℃	0℃~50℃	-20℃~60℃

This module is an excellent choice for our application. The ESP32 can be easily programmed with the well-known Arduino IDE. A micro SD card slot is provided, and the GPS module can be easily connected to the second serial port (first serial port is for Arduino connection). Only a few solder joints need to be made to connect the GPS module:

1. GPS-ground must go to GND
2. GPS-3.3VDC goes to 3 parallel outputs : 25, 26 and 27 (except T5 2.4, no 25).
3. GPS Rx goes to 33
4. GPS Tx goes to 32
5. A reed switch, or button switch,, comes between GND and 39.
6. A optional(reed) switch between GND and GPIO 12
7. Lipo is connected to VBAT and GND (can also be connected via plug).



GPS module ublox M8n engine

Because the logger uses the compact “ubx” protocol, only GPS modules that support it can be used. Modules with the ublox M8n engine are excellent. Some of these cheaper modules don't have flash to store their settings, but this isn't a problem. At every start, the desired setting is transferred again to the GPS module:

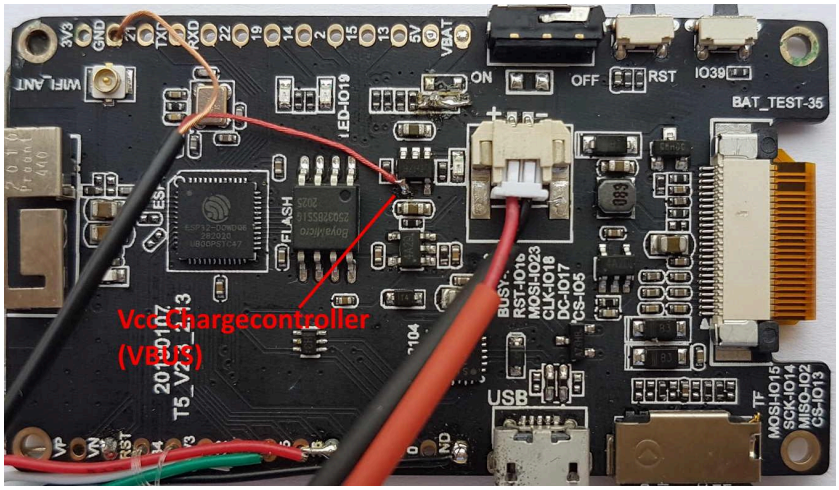
- Serial baud rate is set from 9600 baud to 19200 baud.
- NMEA protocols are disabled.
- UBX NAV PVT protocol is switched on.
- The desired sample rate is set.
- GNSS is set (default GPS+ GLONAS, optional GPS, GLONAS, BEIDOU and GALILEO)

The Beitian BN180, BN220 or the BN280 are both suitable (M8 chip). The BE180, BE220 or the BE280 have the M10 chip, which is even better. The BN280 has a significantly larger antenna. The “first fix” of the 280 is therefore faster than the 220/180. Make sure you have the Beitian branded modules, as they perform a lot better then the “unbranded” ! Please note, the baud rate must be set to 9600/38400 baud by default, since the initialization starts with 9600/38400 baud (at boot) baudrate.. The current consumption of this GPS is

approximately 50 mA(for the M8 / M9, the M10 has 25 mA).. Because the outputs of the ESP32 are allowed to supply a maximum of 20 mA, 3 outputs are connected in parallel. The T5 2.4 version has no GPIO25, but it seems that with 2 outputs (26 and 27) everything works fine.

Lipo battery with protection + charging circuit

Because the TTGO has a built-in charge controller, in principle any lipo battery can be used. However, it is safer to use a lipo with built-in protection. The protection ensures that the lipo cannot overcharge, discharge too deeply or deliver too much current. Because the esp32 is not completely switched off, a small current of about 1.5 mA continues to flow even in the “sleep” position. If the blue LED is removed from the PCB, this current can drop even further to 0.6 mA. With 1.5 mA it still takes more than 1000 hours to fully discharge the lipo. The protection will then switch on and disconnect the lipo, so that it will not be damaged by undervoltage. Charging starts automatically when the micro USB receives power. A red LED indicates charging (only with the older BN versions). The wireless charging circuit (inductive charging) must be soldered directly to the Vcc pin of the charge controller (see photo).



Configuration file

At startup, an attempt is made to read the file “config.txt”. Attention, this file may not contain extra (hidden) characters, these can prevent decoding. The file contains a number of user-configurable settings.

```
{
  "cal_bat": 1.75,
  "cal_speed": 3.6,
  "sample_rate": 2,
  "gnss": 3,
```

```
  "logSBP": 1,
  "logUBX": 1,
  "Speed_field": 1,
  "dynamic_model": 0 ,
  "bar_length":1852,
  "Stat_screens":23,
  "GPIO12_screens":149,
  "Logo_choice":34,
  "Sleep_off_screen":11,
  "stat_speed":1,
  "timezone": 1,
  "Sleep_info":"Call +32 502 003 004",
  "UBXfile":"BN220A",
  "ssid": "SSID",
  "password": "password"
}
```


- cal_bat is the calibration of the battery voltage measurement (1.7 - 1.9).
- cal_speed is for the conversion from gps unit m/s to km/h (3.6) or knots(1.94).
- sample_rate can be 1,2,5,10 or 20 Hz (only M9 chip). The higher, the more accurate, but also the larger the files become! One UBX NavPVT message is 100 byte, so at 1 Hz this gives a file of 360 kb/hour, at 10 Hz 3.6 Mb/hour !
- gnss = 3 : GPS + GLONAS + GALILEO (only possible with ROM version 3.01 !!). Default is only GPS + GLONAS used. Some Beitian modules still have a old firmware, ROM 2.01. Here, Galileo can't be activated. The M9 module can use 4 GNSS simultan, but only with a max rate of 5 Hz (GPS + GLONAS + GALILEO + BEIDOU). The M10 will have "lost data points" with 3 GNSS@10Hz. So, if you need 10Hz, choose 2 GNSS.
- logSBP : To save the GPS data in sbp format. This is also the file that you can upload to gp3s. The sbp format is ca 70% smaller then ubx.
- logGPY : To save the GPS data in gpy format. This is the most condensed format (even 80% smaller then ubx). This is also the file that you can upload to gp3s.
- logUBX : To save the GPS data in ubx format. This is also the file that you can upload to gp3s.
- logGPX : To save the GPS data in gpx format, always 1Hz data, can be used for Google Earth, video overlay... Not suited for upload to gp3s !
- "File_date_time": Different formats for file naming.

- Speed_field : The preferred value in the upper speed screen : 1=Run, 2=Dis, 3=AVG, 4=2s, 5=Alfa, 6=0,5h, 7=1h, 8=100m. After start-up, the speed_screen can be changed with a short push on the button / reed switch.
- "dynamic_model": Here you can choose the dynamic model of the Ublox M8N (0=portable, 1=sea, 2=automotive). As “sea” has some disadvantages (max speed is limited to 40 knots, only sea-level..), my advice is to stay with “portable”.
- "Stat_speed": The max_speed in m/s to show the stat_screens, default 1 m/s
- “Start_logging_speed”: At this speed, the logging will start.
- "timezone": The local time difference in hours with UTC (can be negative !)
- Sleep_info : Text appears in sleep_screen.
- UBXfile : Here you can set the desired file name, this is completed with the (unique) MAC address of the ESP32 and a suffix from 000 to 999, or combined with date&time
- ssid : The SSID of your own network.
- password : The corresponding password

With the latest firmware (SW5.50 and higher), the config.txt can be written with the webserver. If there are no config.txt or config_backup.txt, the ESP logger will boot with the default settings. You can have WiFi access with logging in to the Acces Point (SSID= ESP32AP, password = password). The IP is visible on the e-paper : 192.168.4.1.

With a browser, you get access to the logger. Choose “configuration” , and change to the desired settings (your home Wifi SSID and used password). With “SUBMIT”, a new config.txt is written to the SD-card.

If you try to connect to the AP with your mobile phone, you have to switch off “mobile network” ! Because the ESP32 has no internet connection, your phone will try to find internet over the mobile network (4G). This will block the webpage access. For Android, change the settings : **Settings, Connections, Data Usage, Mobile data OFF**

Wifi functionality (2.4 GHz)

When starting up, the configuration file is first read from the micro sd card (config.txt). The logger then tries to connect to the SSID which is set in the config.txt. If successful, the IP address will be shown on the display and the message “Wifi On”. If your computer (or smartphone) is connected to the same Wifi network, you can access the logger over this IP (this is a “local” IP, that is only visible in you own SSID !).

It is also possible to set the ESP32 as an access point. While searching for the set SSID, the reed switch must then be actuated with a magnet, then the ESP32 switches to AP mode for at least 100 seconds. You can log in to the SSID “ESP32” with the password “password”.

As long as there is range, the Wi-Fi will remain active. The wifi connection can be used for 2 things:

1. Establishing an ftp connection (file transfer protocol). Here you can view, download and, if desired, delete all files on the SD card. A common ftp program for Win 10 is

"WinSCP" : https://winscp.net/eng/download.php With Apple OS, there is a issue with ftp, it does not work with the ESP32. With a Iphone, ftp seems to work normal. I did a short test with FTPManager : https://apps.apple.com/us/app/ftpmanager-ftp-sftp-client/id525959186

2. There is also a built-in web server for the access to all files, change of the configuration and "Over the air" updates. (OTA does not work in AP mode!) With a browser you can surf to the indicated IP address. Attention, for changing the firmware you also need the correct update file (bin) ! On github, you can find the latest version for your type of screen (Bin files) : https://github.com/RP6conrad/ESP-GPS-Logger
3. If there is a GPS fix, and no Wifi connection, other screens will become active.

Controls

The screen itself has some controls :

- A slide switch to completely turn off/on the screen. Charging the battery is always possible, so also in the "off" position.
- A "Reset" push button (see fig. "RST"), with which a reset can be performed.
- A push button going to "IO39" will connect input 39 to GND. This allows you to wake up the logger from "sleep". The "reed" contact has exactly the same function, but can be operated with a magnet. Switching off the logger (to "sleep") is also done via this key/reed contact. The contact must then be operated for at least 2s.
- A USB micro connection. This allows the battery to be charged, but it is also a "virtual" serial port. With a suitable program (HTerm, Arduino IDE..) you can read the startup information here (115200 baud).
- A recent version of the T5-epaper has no serial chip anymore, and a USB-C connection. Here you need a additional interface : T-U2T !
- A Blue LED, it lights up as long as the screen has power. Attention, with the new version this LED is no longer present!
- A Red LED, it lights up when the battery is being charged. This LED is also no longer present in the newer version.
- A plug to which the lipo can be connected. The battery can also be soldered directly to the pin VBAT and GND.

Operation

A Micro SD card is required for logging. The config.txt file must then be placed on this SD card. This SD card has to be formatted in fat32. Some high capacity SD-cards will not work, the Sandisk Ultra Plus 32GB will not work ! Best to stay with max 16 GB cards. More recent, I had problems with "unreliable" sd-cards, even with only 16 GB or less ! They work sometimes. So test your sd-card in the ESP logger with high load (10 Hz, writing .gpy, .ubx at the same time).

Switching On

Switch on from “sleep” can be done by operating the “reed” contact with a magnet. This is possible with a closed housing.

At startup, the config file is first read, then an attempt is made to connect to the set SSID. If successful, a screen with “Wifi ON” and the assigned IP address will appear. If the network in question is not found, the Wi-Fi will be turned off. Next text : “Wifi OFF”. The GPS is always turned on and immediately configured.

Start up screen

SD card OK, config info

Information in start up screens :

SW-version 5.48 : The software version
SD OK : SD-card detected Port : dynamic model (can set to “SEA”, “Portable” and “Auto”)
GoPro9 : File name (is set in the config.txt)
Rate : Sample rate, can be set to 1, 2, 5 or 10 Hz.
Speed : km/h or knots, can be set in the config
Config OK : Configuration file is read without failures

Wifi network found

If a connection can be made with the set SSID within 10 s, and there is no GPS fix yet, the following screen appears:
Wifi on ST ! : Wifi connection in station mode (SSID from config).

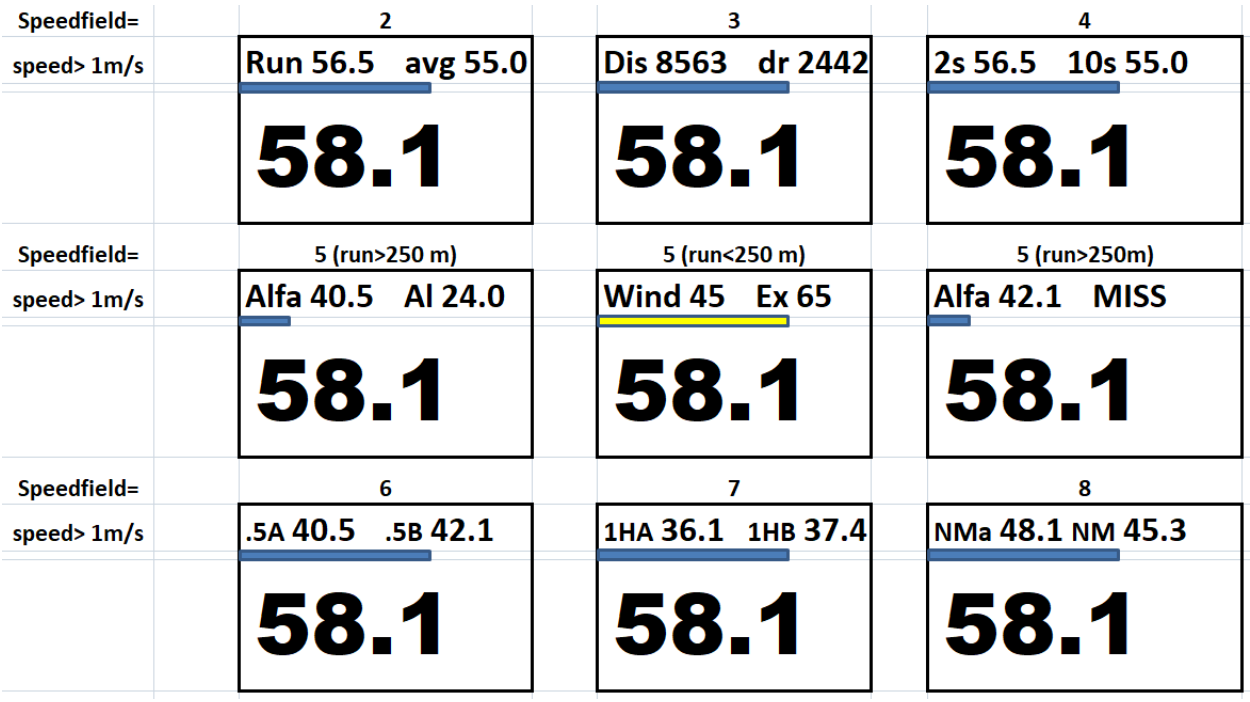
192.168.0.234 : Local IP adres
FTP status : 2 FTP server started, but no active connection
FTP status : 5 FTP connection active
Bat: 3.91 Bat voltage, completely charged is 4.2 V, empty is 3.2 V

Normal operation

As long the Wifi connection is active (Wifi on), you get the “Wifi screen”. The gps is powered, but logging is not started.

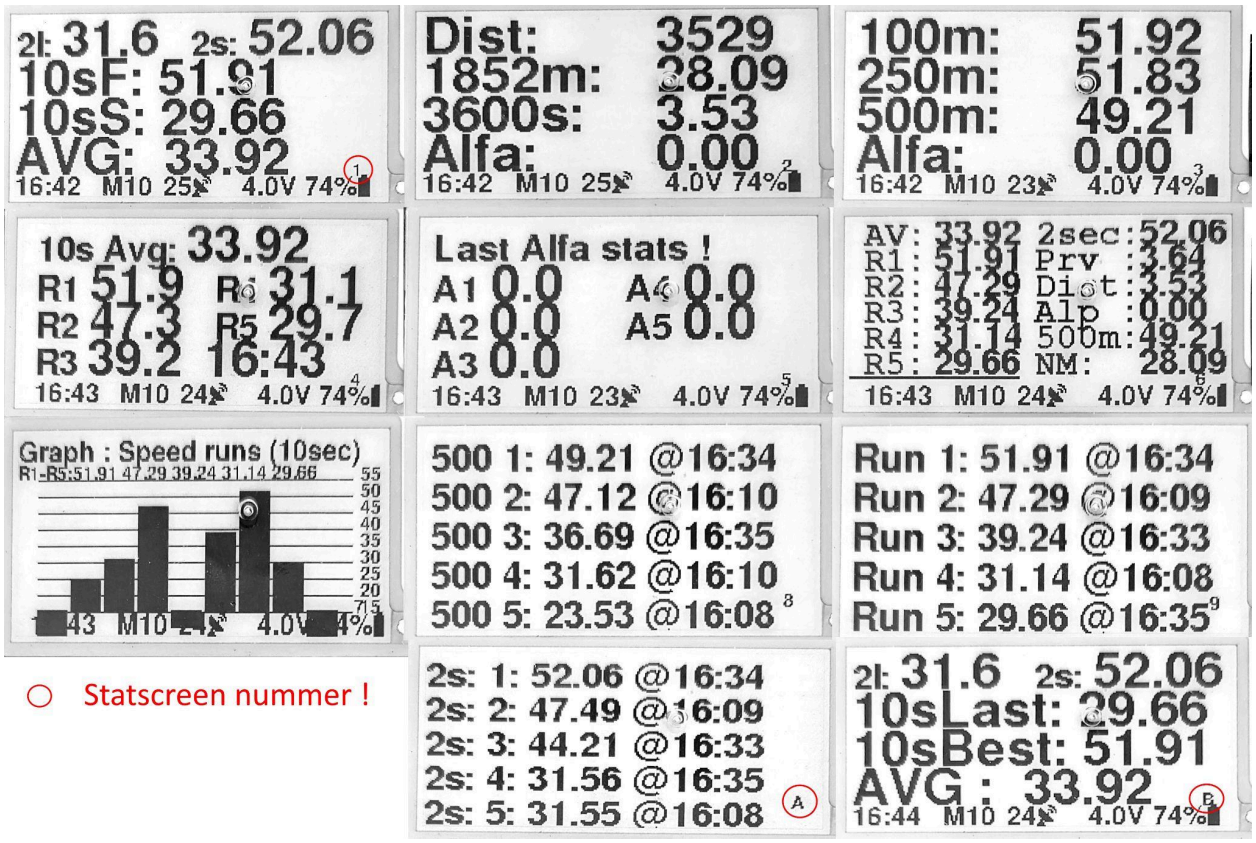
If the Wifi is not active, and a fix is found (approx. 30s - 100s), logging is started as soon as minimal speed is reached (2m/s) in the ubx and oao format. When stationary, the “stat” screens will toggle. As soon as the GPS doppler speed exceeds 1m/s, the “speed” screen appears with the current speed in a big font, and other information :

Speed screens :



If the reed-switch is briefly actuated, the upper field (Run) can be changed to total Distance, 5*10s AVG, best 2s, Alfa, 0.5h avg, 1h avg, best 100 m. Preferred choice can also be set in the config.txt file.

SW6.01 stat screens :



○ Statscreen nummer !

Switching off

The logger can be switched off ("sleep") by pressing a magnet on the “reed” contact or button for at least 2 s. A txt file is then saved with all the results. The naming of the files is as follows:

- The first part comes from the config.txt "UBXfile":"BN280A", here then BN280A
- The second part is the (unique) MAC address of the ESP32
- The third part is an index of 000 to 999
- The last part is then .ubx, .oao and .txt
- Ex: BN280A_ 83AF2466E48_001.txt
- The date and timestamp comes from the gps fix, this is UTC time. Via the configuration, one can set a offset for the local time (eg Belgium summer time = +2). If your date has a 1 day offset, you can use a “negative” offset ! Attention, the FTP server sometimes dares to adjust the time of the files to “local” time!

Sleep-screen 0

Sleep-screen “0” shows the DIST (distance) in meters, the 5*10s average, and the 2s values from last session. The current consumption now drops to 1.5 mA (LED) or 0.6 mA (without

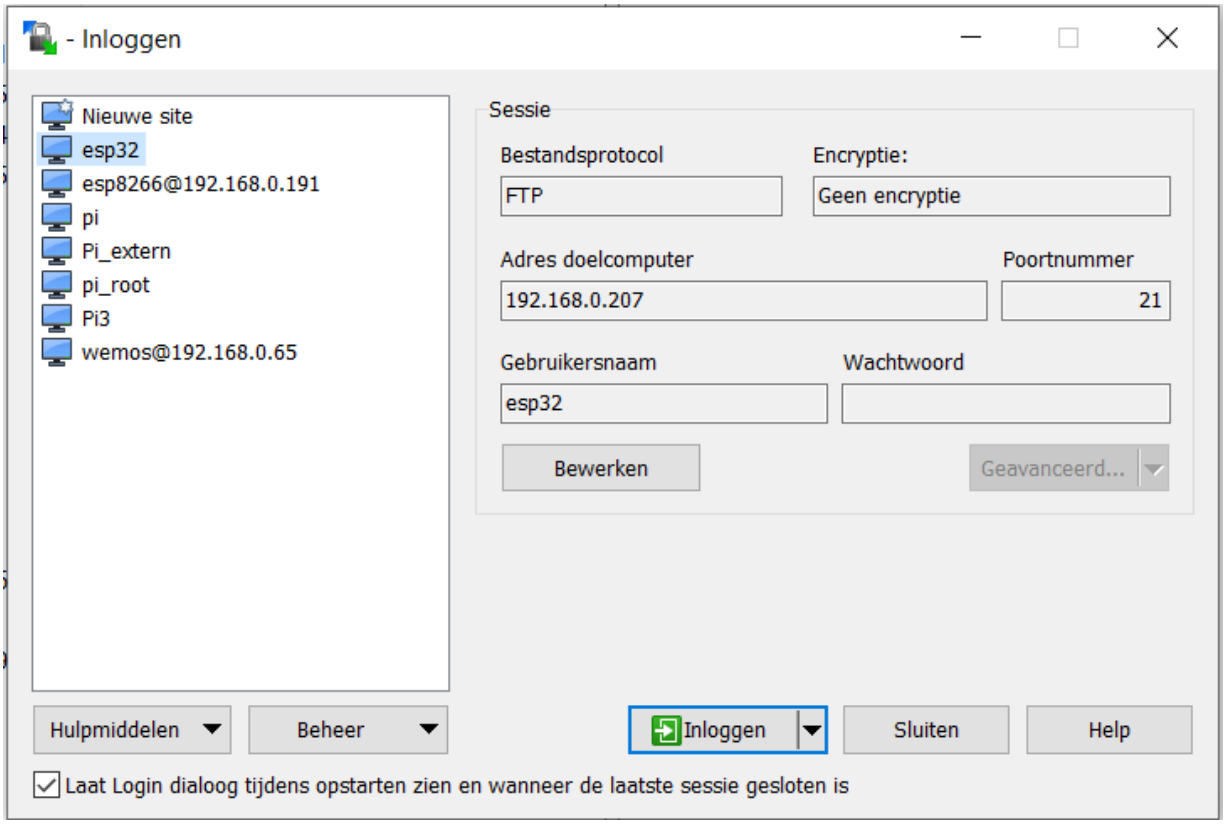
LED). The logger wakes up every hour to measure the battery voltage again. The normal range is from 4.2 (full charge) to 3.2 (empty). If the battery is completely discharged, the protection in the lipo battery will cut off the voltage completely.

Sleep-screen 1

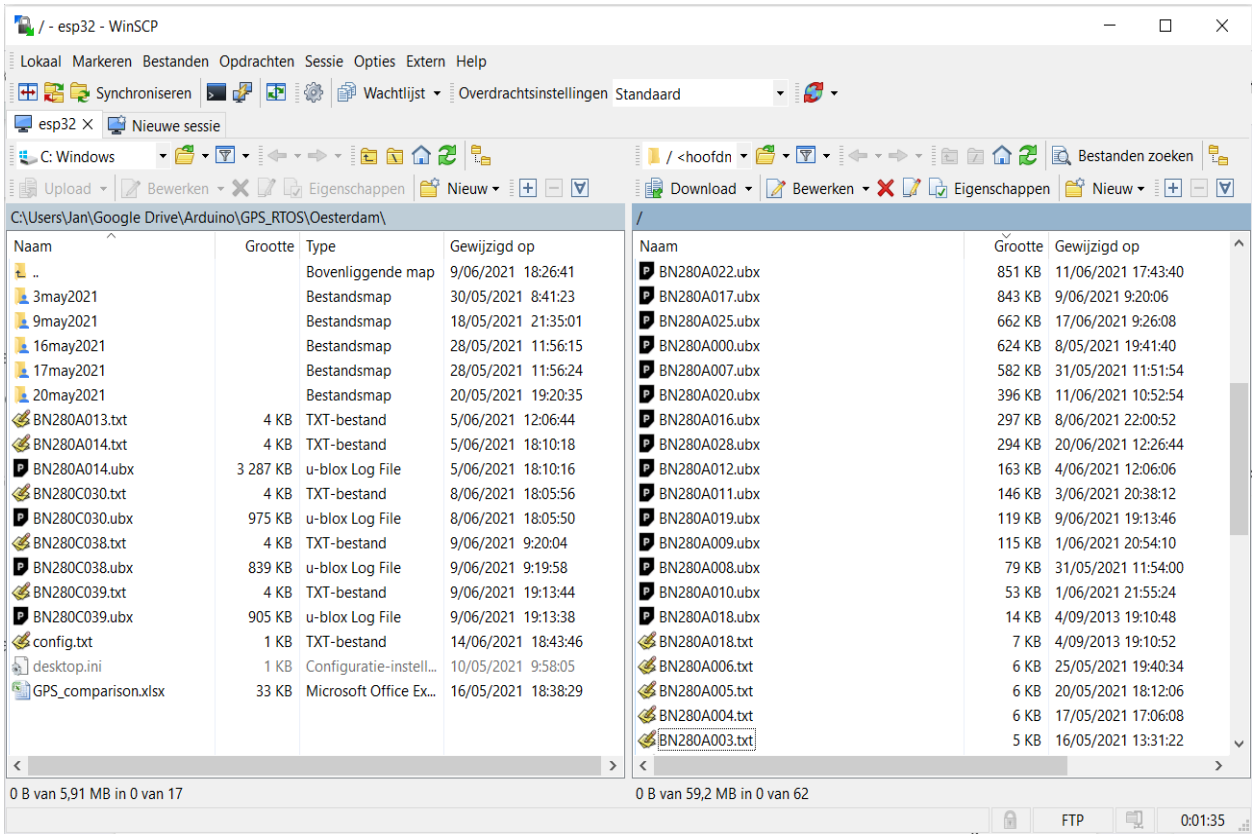
Copying files with FTPcopy

You can always copy the files from the micro SD card directly, but then the housing must of course be opened. The wireless option is via “FTP” (file transfer protocol). The WiFi connection must of course be working for this. You also need to install an “FTP client” on the PC. A commonly used FTP client under Window is “WinSCP”. The procedure is as follows:

- Start the logger and wait for the Wifi connection (ST or AP mode)
- Start up the FTP program, settings “FTP”, user “ESP32”, port 21 and enter the IP address of the logger.
- Log in with the user “esp32” and the password “esp32”
- You will now get an overview of the files on the esp32.
- Copy the desired files to the PC via “drag and drop”
- If desired, you can also delete the files on the logger afterwards.



After logging in, you will then see the following screen with the “remote” files on the logger on the right, and the “local” files on the PC on the left. With drag and drop the files can now be copied. The “.ubx” file contains the GPS tracks, this file can be uploaded to GP3S. The “.txt” file contains all the speeds calculated by the logger itself.



FTP with your smartphone

I had good experience with FTP Manager (Iphone) and FTP Cafe (Android). Make sure you use the setting flat FTP (without TLS).

FTP on MacOS

Here, FTP can be used with “Filezilla”. When you try to transfer a file, it seems that nothings happen, but after 20s, the download is starting. Be patient !

Lipo battery charging

Charging the lipo is easy via the micro USB connection. The T5 board has an integrated charging circuit, which ensures the correct current and charging voltage. It is charged with about 500 mA, so the 2000 mAh lipo can be charged in 4 hours. If an inductive charging circuit is provided, wireless charging is also possible. The coil is placed on the bottom of the box, over which the ferrite adhesive strip goes.

This wireless charging works well, but because the distance between the coils is quite large, it has a very poor efficiency. Charging can be interrupted by heating (charger has a temp.

sensor and switches off), after cooling it starts charging again.

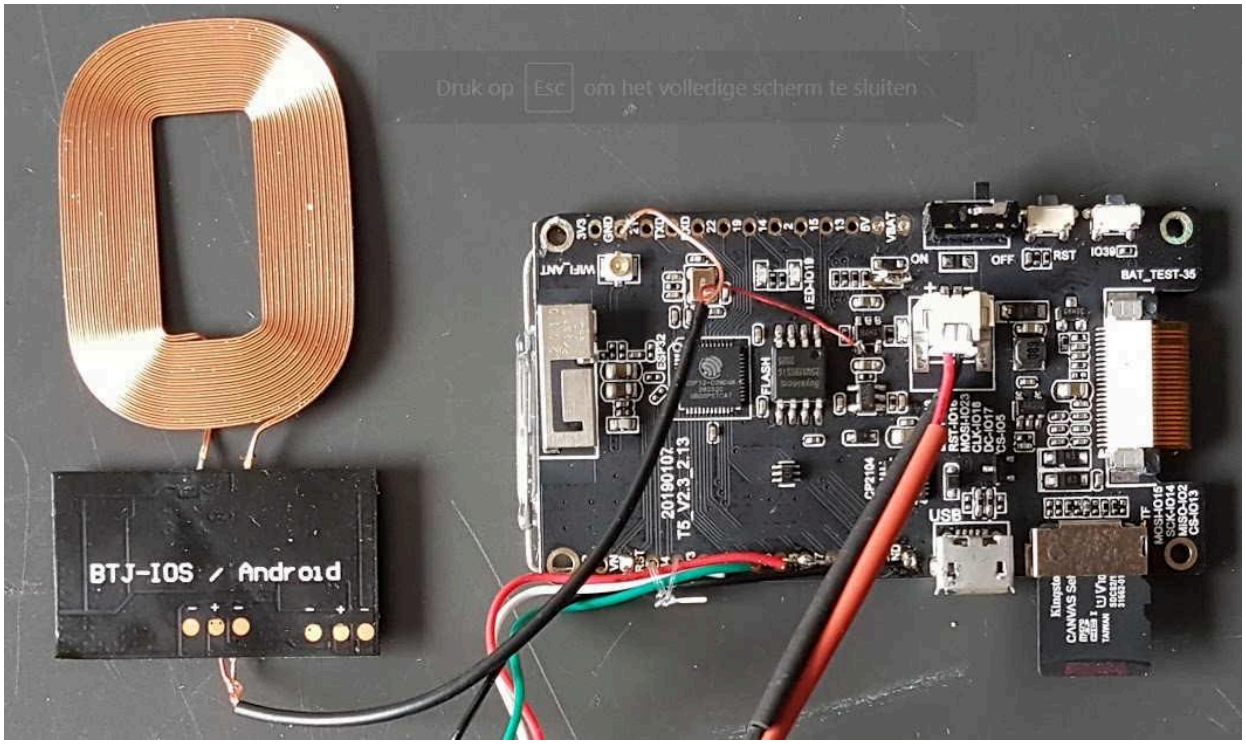
Warning....Wireless charging!!!

When in our case a esp-gps is fully assembled we use only a 5watt charger (5v 1amp) like the (old) orginal iPhone chargers. This way the produced heat is limited. As i test the hardware before assembling, i recently used a 20watt apple charger (usb-c). should be no problem because the wireless charger and reciever is not in a housing so the heat can go away.

So much for the theory, in practise the wireless reciever burnsout on the end of the charging. First one was a 213BN (no chip with leds) usb-c charger reciever.

At first I thought it was an exemplary error, so i charged a B74 with the micro-usb reciever. Hmm same result. Burnsout at the end of the charging.

The 20 watt apple charges has 9volt 2.22 amp or 5volt, 3amp.



Loading new firmware

Via the WiFi connection you can easily upload new firmware (OTA, over the air update). The correct “bin” file is of course necessary! Attention, there are different versions of the T5 (each time a different epaper), so you need the bin file that belongs to the screen.The sw is now completely open source, all files can be found on github :

https://github.com/RP6conrad/ESP-GPS-Logger

If the WiFi connection is active, you can simply surf to the indicated local IP address with a browser. A login screen will then appear where the user “Admin” and the password “Admin” give access to the upload screen. Now simply select the correct “bin” file and start the

upload. After about 20 seconds the file has been transferred, and the logger restarts with the new firmware.

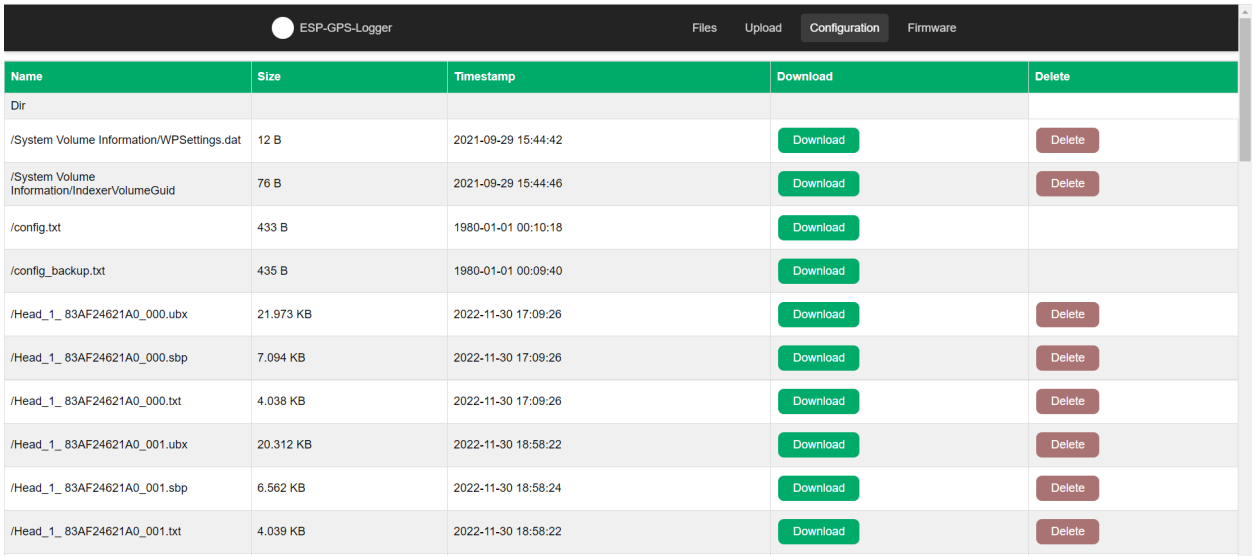
Webserver functionality, from SW 5.56 !!

As the project is open source, Tritondm (github) has developed a beautiful webserver for the ESP-GPS, so every interaction can now be done over the webserver ! No need for FTP anymore. Even the settings in the config.txt file can be done over this webserver !

This webserver will work in SSID and AP mode. If you try to connect to the AP with your mobile phone, you have to switch off “mobile network” ! Because the ESP32 has no internet connection, your phone will try to find internet over the mobile network (4G). This will block the webpage access. For Android, change the settings : **Settings, Connections, Data Usage, Mobile data OFF**

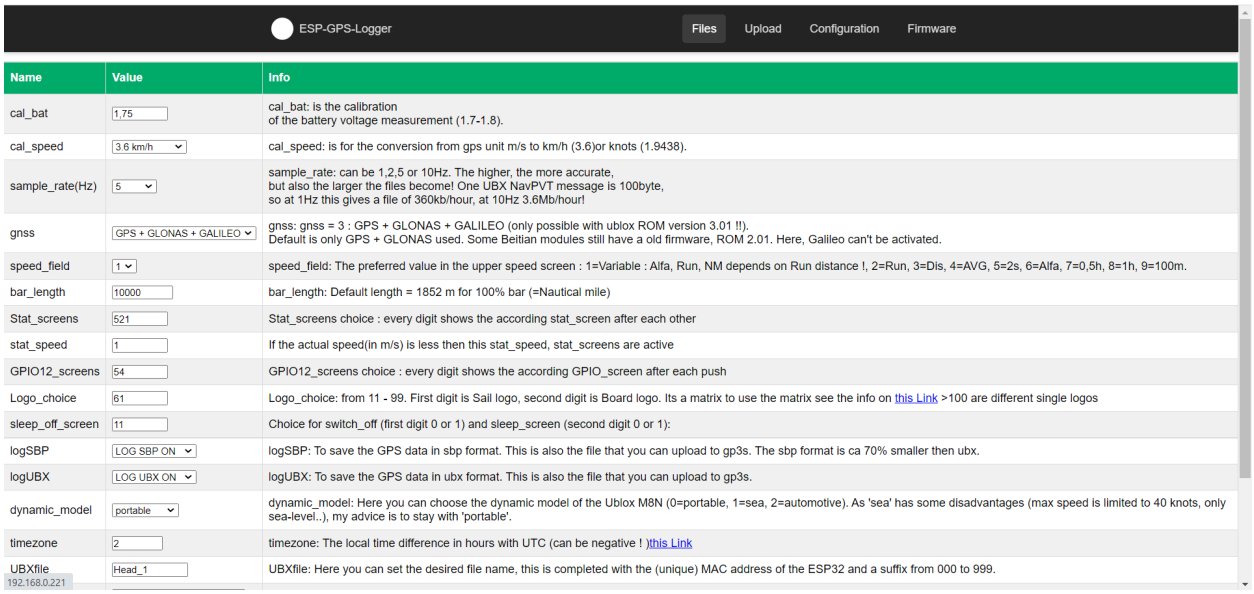
Some screenshots if you open the webserver (just over the IP adres) :

Mainscreen : To download your session files (.ubx or .sbp)



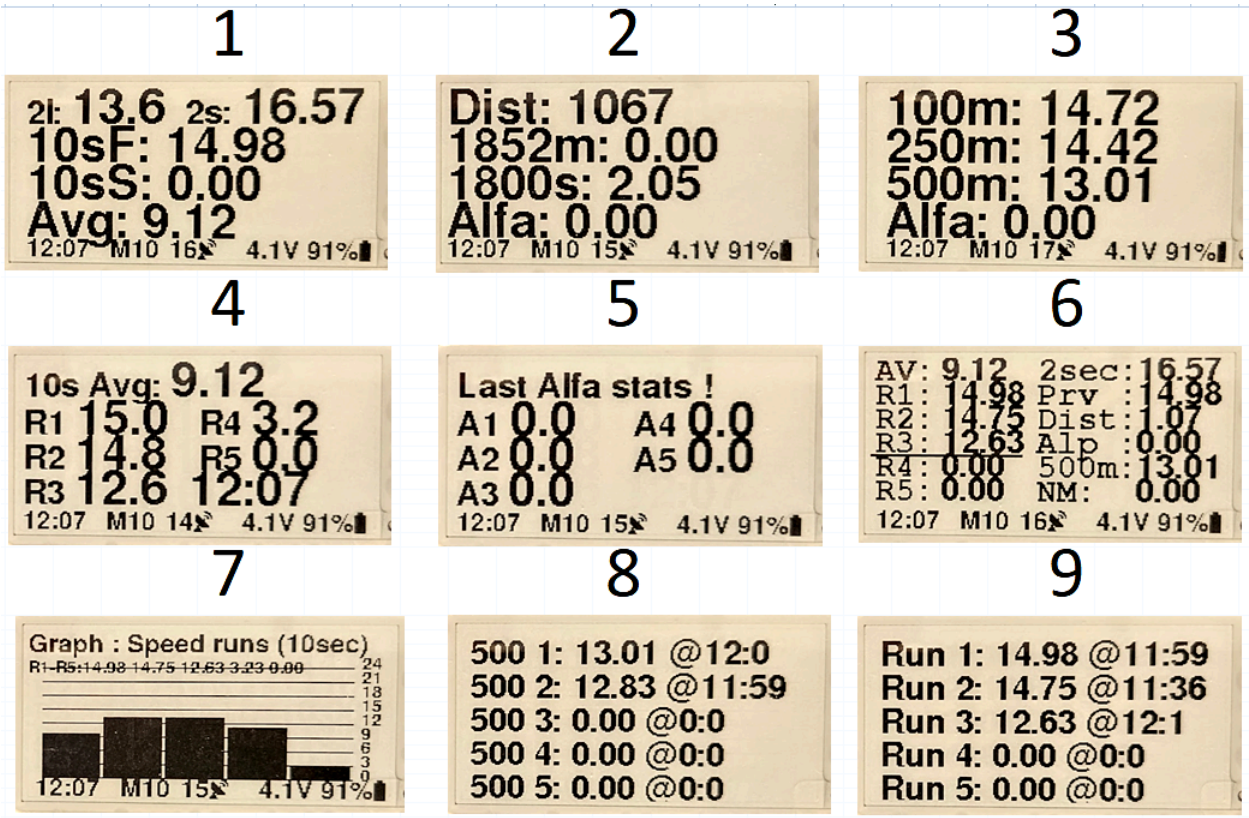
Name	Size	Timestamp	Download	Delete
Dir				
/System Volume Information/WPSettings.dat	12 B	2021-09-29 15:44:42	Download	Delete
/System Volume Information/IndexerVolumeGuid	76 B	2021-09-29 15:44:46	Download	Delete
/config.txt	433 B	1980-01-01 00:10:18	Download	
/config_backup.txt	435 B	1980-01-01 00:09:40	Download	
/Head_1_ 83AF24621A0_000.ubx	21.973 KB	2022-11-30 17:09:26	Download	Delete
/Head_1_ 83AF24621A0_000.sbp	7.094 KB	2022-11-30 17:09:26	Download	Delete
/Head_1_ 83AF24621A0_000.txt	4.038 KB	2022-11-30 17:09:26	Download	Delete
/Head_1_ 83AF24621A0_001.ubx	20.312 KB	2022-11-30 18:58:22	Download	Delete
/Head_1_ 83AF24621A0_001.sbp	6.562 KB	2022-11-30 18:58:24	Download	Delete
/Head_1_ 83AF24621A0_001.txt	4.039 KB	2022-11-30 18:58:22	Download	Delete

Configuration screen :



Statistic screens (speed<1m/s)

There are 10 stat screens, which you can choose in the config file with the parameter “Stat_screens”. If you set 321A, the screen 1, 2 ,3 and A will toggle every 2s (time configurable). You can set a max of 9 screens. If you set the stat screens time to 0, you can toggle the screen with a short push of GPIO39 (on/off reed switch or button, SW6.01).



Exactly the same with the “GPIO_12_screens. You can set the desired screens in the same way. If the GPIO_12 is activated, the screen will be visible for ca 10s. With every short toggle, you switch to the next screen. With a long push, you can reset the runs and alfas in screen 9 and 5.

SW 5.4 : some changes in the speed screens (speed>1m/s) :