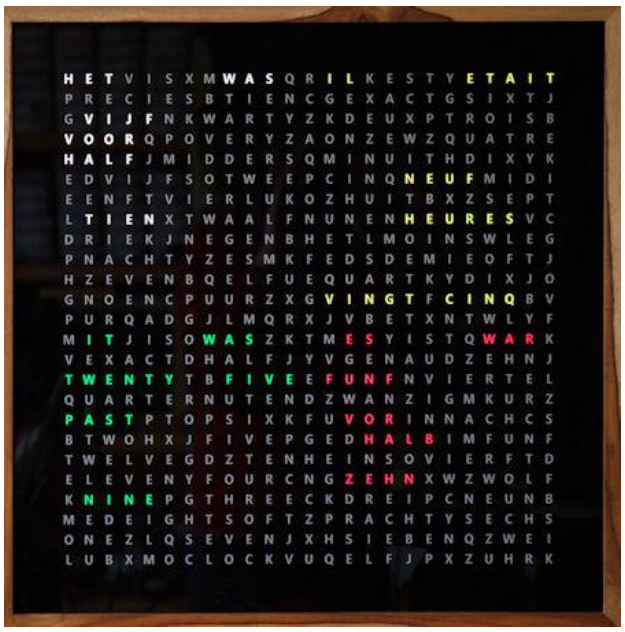
# Arduino ESP32-Nano word clock

A clock that displays the time in words in the languages Dutch, English, French and German in a large 4-language clock or as a monolingual clock.

An Arduino ESP32 Nano computer in the clock is used to control the clock and to establish a connection to WIFI and Bluetooth.
The time is synchronised with the Internet's Network Time Protocol (NTP) via a WIFI connection. In addition, there is the option to set the time without WIFI or Bluetooth with a turn/push button.
Settings can be managed using a web page, a PC, or a Bluetooth Low Energy (BLE) serial terminal app installed on a phone, PC, or tablet.




4-language clock


Dutch language clock

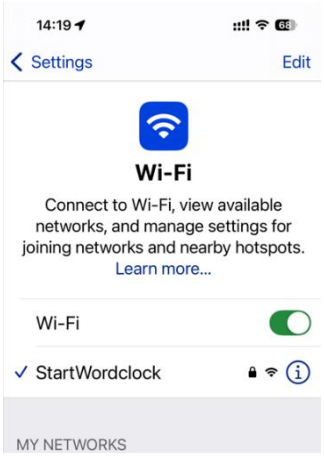

Corten steel clock

# Quick start

Find one of the following apps in the app store.
For **IOS/MacOS:** BLEserial or BLEserial Pro and
for **Android** Serial Bluetooth terminal. (Turn off timestamp in this app's menu.)

| IOS iPhone/iPad/Mac | IOS iPhone/iPad/Mac | Android |
|---|---|---|
| BLE Serial Pro | BLESerial nRF | Serial Bluetooth Terminal |

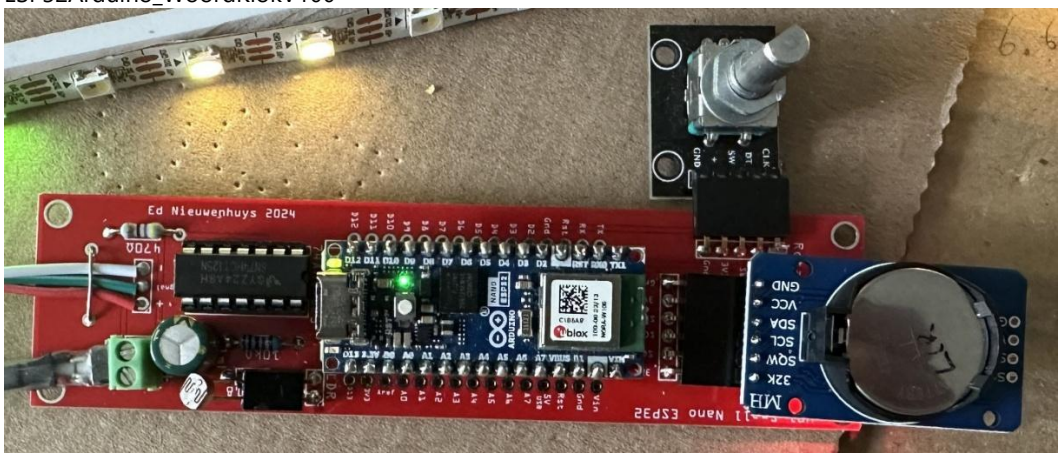
By default, the clock is set to receive the time with WIFI.
If the name of the WIFI router and password are not yet set a web page is started to enter the credentials.
Open the WIFI-station "StartWordclock" in a phone or tablet in the WIFI settings.
Enter the password: wordclock
then open in a browser on the mobile and type in the URL: 162.168.4.1 and press enter. Enter the name and password of the WIFI router used in your home. (KPN modem or something similar.)




Setting up the WIFI data is easier with the **Bluetooth app** on the phone or tablet.  Open the app.
The IOS app shows a list of Bluetooth stations. If it takes too long. Press wheel at the top left (nRF52 is selected) and press Done. In Android app, press Scan and Connect with the word clock.
Send the command: **i**
A menu follows. (Do not send extra points or spaces)
Send the character **a** followed by the name of the router and press Send.  (e.g.: aNAMErouter)
Send the character **b** followed by the password of the router and press Send.  (e.g.: bSeCret1234).
Send @ to restart the clock.


**Or start WPS** (A button on the router with WPS that you press. Sometimes for a few seconds. (Read this in the manual).
Press the WPS button and then send the command: **z** in the Bluetooth serial app

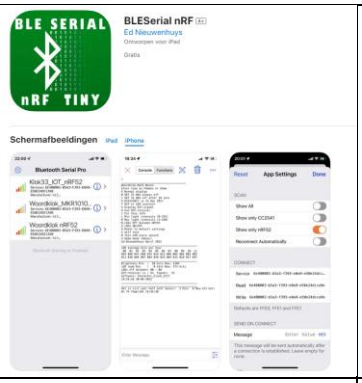
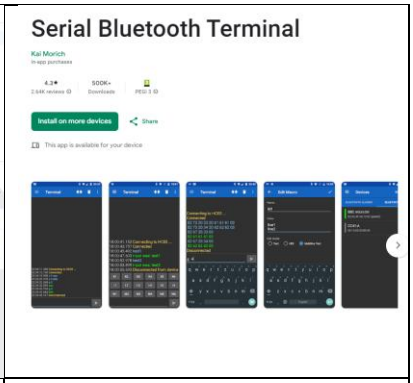
**If no WIFI is possible**
Send the letter J in the menu with the Bluetooth app to start the accurate internal clock. The app will display: Use DS3231 is ON. WIFI ON, NTP OFF
Send the letter W to turn off WIFI. (if you don't use the WIFI menu.)
Send @ to restart the clock. Set time and date in the app menu. (e.g. send: T143000 before half past three and date: D250625)



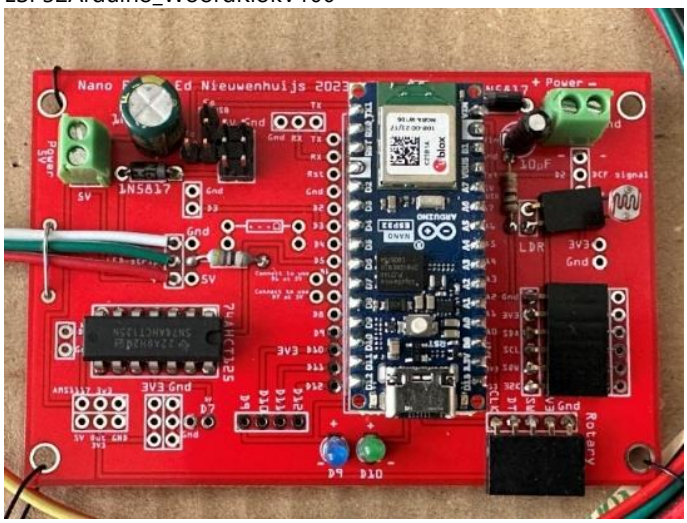

Small PCB design with rotary dial and DS3231 RTC attached

## Before starting

The clock receives the time from the internet if there is a WIFI connection.
When a DS3231 time module is attached to the circuit board, an internet connection is not required.
If a rotary or push-button button is installed, it can be used to set the time.
To connect to the internet, the name of the WIFI station and the password must be entered into the clock software in order to connect to a WIFI router.
The name of the WIFI station and the password has to be entered once. The credentials are stored in the microprocessor's memory.
A serial terminal app on a phone, tablet or PC allows operation without connecting the clock to a PC with a USB cable to enter the WIFI login details.
The clock can also be controlled via a browser if a connection is made to WIFI.

| | | |
|---|---|---|
|  |  |  |
| BLESerial nRF | BLE Serial Pro | Serial Bluetooth terminal |

- Download a Bluetooth UART serial terminal app on a phone, PC or tablet.
For IOS: BLE Serial Pro or BLESerial nRF.
For Android: Serial Bluetooth terminal.
For a PC: install the Arduino IDE on a PC and connect the clock to the PC via a USB-C cable.




Arduino ESP32 Nano on the circuit board in the clock.

## First Operation

- Connect the clock's power supply.

**If there is no internet connection or Bluetooth connection**, the clock can be operated with the rotary knob or with three push buttons, provided it is installed and activated. Activation is done in the menu with option H001 (rotary knob) or H002 Membrane knob.  Sometimes the knob is mounted in the cabinet.

- Press the rotary shaft or the middle button of the three push buttons.
  UUR (or Oclock)  lights up 3 times and the hours can be set by turning the knob or pressing the up or down button.
- A second press of the button or the middle button will flash
  HETISWAS and the minutes can be changed.
  Seconds are set to 0. So to be precise, set the last minute to exactly 0 seconds.
- With a third press, TWELVE flashes three times and the light intensity of the LEDs can be adjusted.
- A fourth press will flash the characters NTP and RTC three times. One can switch from RTC to NTP time.
- Nine presses reset the clock to factory defaults.

**Via WIFI connection**

When the clock is started for the first time, a WIFI connection will have to be made.
If there are still old connection data in the clock, enter 5 times capital R in the menu (RRRRR) to delete them.

The clock settings will then be set to:
SSID and password are empty
Timezone:CET-1CEST,M3.5.0,M10.5.0/3
WIFI=On NTP=On BLE=On

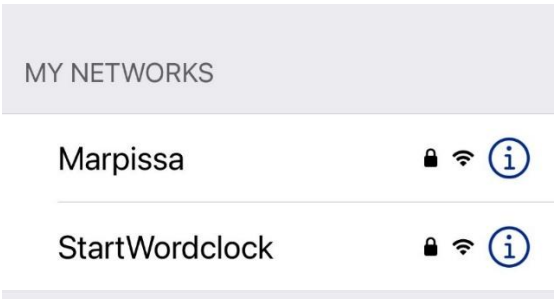
**WPS**

WPS is a method that can be used to receive the credentials from the router.
Start WPS in the clock by entering Z in the menu and start WPS on the WIFI router. If all goes well, the clock will restart and the time will be the same.

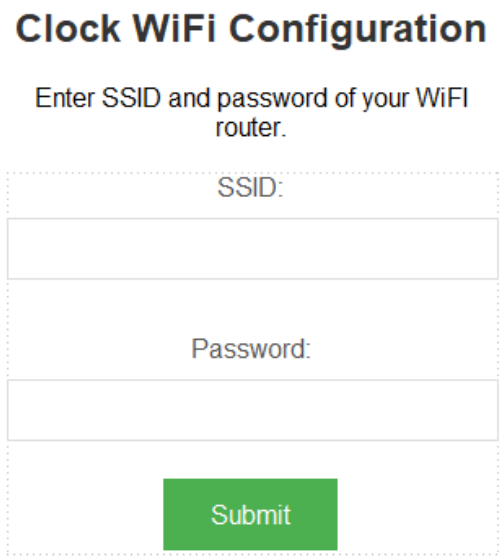
**Access point in the phone**

With the WIFI connections in the phone (tablet on PC) there will be a "StartWordclock" station at the router you are connected to.

- Connect to the "StartWordclock" station. Use as password: **wordclock**
- Open a browser and type: 192.168.4.1 and press enter.




- Type in the SSID and password of the WIFI router in this screen and press Submit
  This data is often located at the bottom of the router.




The clock will restart and display the correct time.

**When the Bluetooth app is installed on the phone or tablet**

With the 'Bluetooth serial terminal' app, the clock can be controlled on your phone, tablet of MacOs PC via Bluetooth.

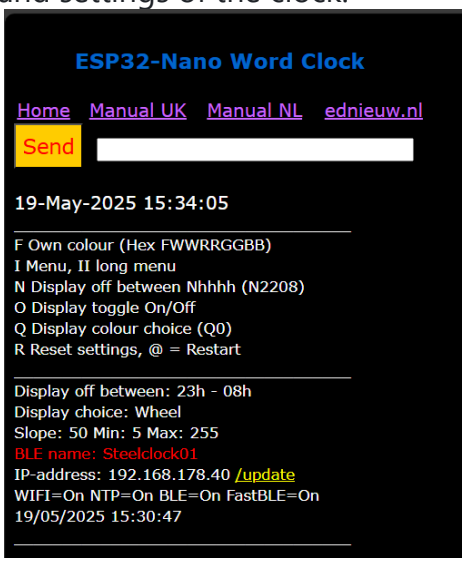
- Launch the Bluetooth terminal app on the phone or tablet
- Click on the connection called "wordclock" in the serial terminal app.
- Find your WIFI router's credentials.
  Find the router's SSID WIFI and password. These are normally located on the bottom of the WIFI router

Type in:

- aSSID and press Send.
  SSID is the name of the SSID that you can find on your phone, for example. It is the name that WIFI is connected to.
- bPASSWORD and hit send.
  PASSWORD is the password that you can find on the bottom of the WIFI router.
- cBLENAME (optional, otherwise the clock will be named 'Wordclock')
  The BLENAME is the name of the clock in the BLE serial terminal app and in the connected WIFI router.

If the Router name (SSID) or password is not correct, the clock will start the Access Point page (192.168.4.1 ) in the phone again after a few reboots.
Via this access point page, Bluetooth or with the USB cable the correct data can be entered.

For a detailed explanation, see chapter: Operation and settings of the clock.


In the FRITZBox WIFI router, the clock named "SteelClock-01" is connected to WIFI as an example.
You can start the clock's webpage in a browser by tapping:
http://SteelClock-01 or SteelClock-01.local.
The default name of the clock is wordclock. De URL to start the menu page will then be: wordclock.local

With the option I in the menu, you can choose between a short menu for everyday use or a long menu with option II (2x i) for an extensive menu with all options.

## Construction of the clock case

The construction of the clock case with lighting can be found here:
https://ednieuw.home.xs4all.nl/Woordklok/Bouwpakket/WoordklokSK6812.htm

or in this repository
https://github.com/ednieuw/Arduino-ESP32-Nano-Wordclock

Building instructions of a 4-language word clock with SK6812 LEDs in UK, NL, DE, FR with Nano. https://github.com/ednieuw/FourLanguageClock

A rotary encoder and DS3231 RTC are optional and not necessarily necessary. They are necessary if a clock cannot receive WIFI.
In the clock menu, the rotary encoder (or three push buttons) can be turned on or off. Turn off the option if no rotary encoder is installed.

## Compilation and uploading

The software runs on an Arduino Nano ESP32.
This software is compiled with the Arduino IDE and uploaded to the Nano ESP32.
There are also .bin files available.
A compiler is available for every board that can be programmed with the Arduino IDE. This can be selected in the boards manager menu
There are two versions available for the Arduino Nano ESP32.
One version, from Arduino itself, which uses core version 2. This version is easy to use.

## Optional

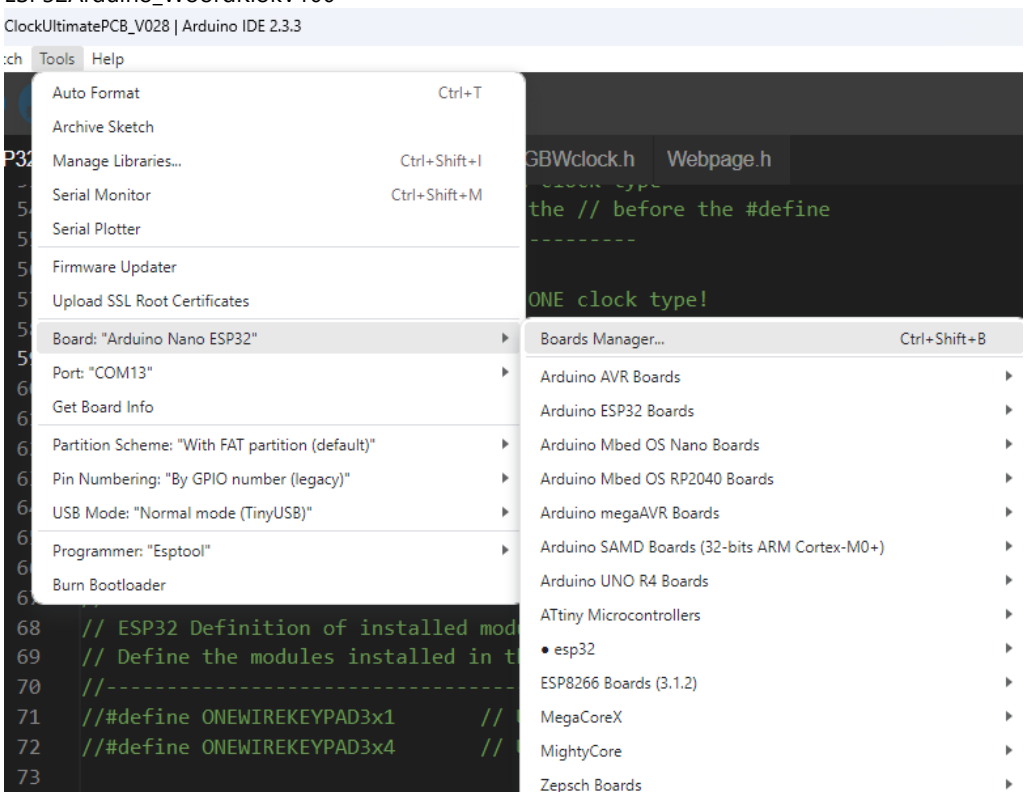
The manufacturer of the ESP32 microcontroller Espressif is developing core version 3.
Since version 3.0.5, this core compiled the sketch "ESP32Arduino_WordClockV057" without any problems.
**Unfortunately with version 3.1 not anymore.**

- Search for ESP32 in the board manager.

The settings of the Arduino Nano ESP32 board are as follows:

- Install ESP32 boards.
  The Arduino Nano ESP32 can be found somewhere at the bottom of that long list of boards if the Espressif core V3 is used.

ESP32Arduino_WoordKlokV100




- Load the ESP32Arduino_WordClockV0xx.INO file into the IDE

In the sketch, somewhere around line 50, select one of the three word clocks by removing the two slashes //.

```
//#define FOURLANGUAGECLOCK
#define NL144CLOCK
//#define NLM1M2M3M4L114 // NL clock with four extra LEDs for the minutes to light up
```

#define NL144CLOCK -> a 144 LED monolingual clock. The standard language is Dutch. For French, German and English, copy the coding of the quadrilingual clock between the NL144CLOCK definitions.

#define NLM1M2M3M4L114 -> a 110 LED monolingual clock with 4 additional LEDs for the minutes and a slightly different design

#define FOURLANGUAGECLOCK > a 4-language clock with 625 LEDs in a grid of 25 x 25 LEDs.

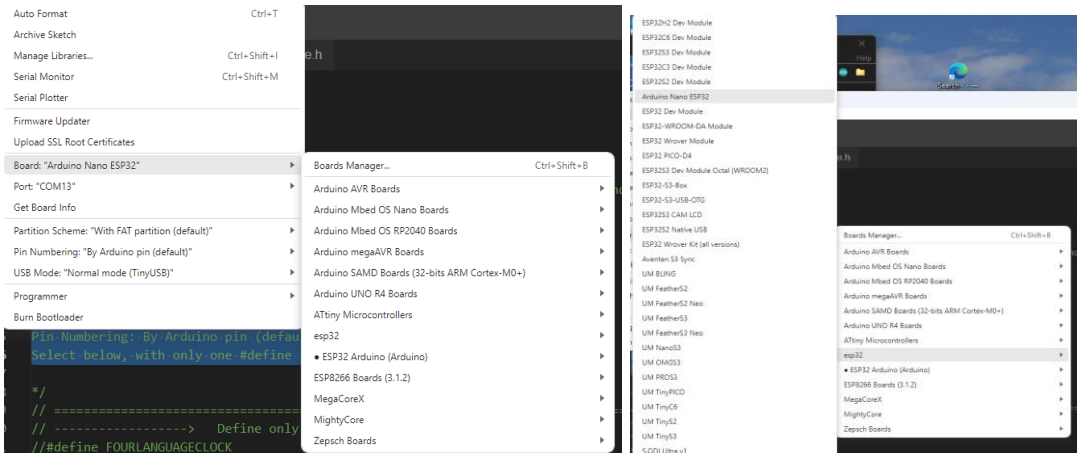
The libraries.zip contains the libraries to compile the software. Unzip them in the libraries folder.  This folder is usually located in the folder with the sketches.

Board: Arduino Nano ESP32
Partition Scheme: With FAT
Pin Numbering: By GPIO Number (Legacy)
(if the Arduino pin (default) is chosen, the LED strip may not turn on and the RGB LED on the Nano ESP32 may not work)

**Problems / happy accidents**
If the LEDs do not turn on, there is a good chance that the pin numbering setting (By GPIO-number (legacy) is set to By Arduino pin (default).

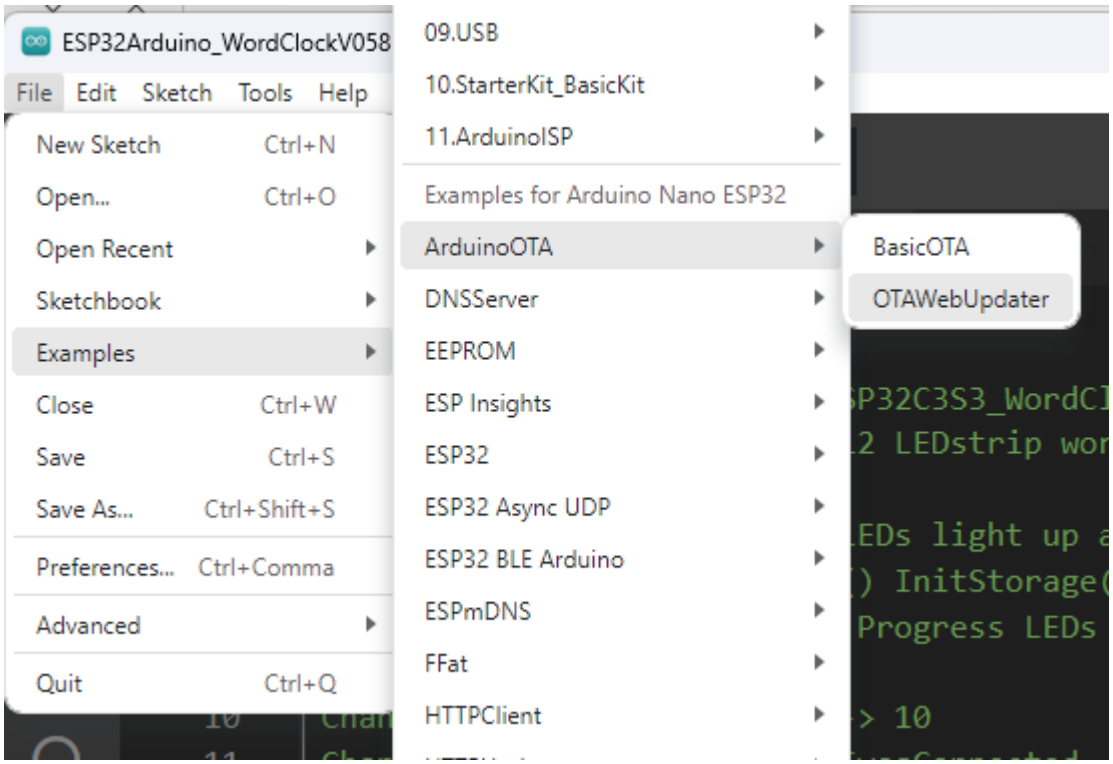
More than one clock is selected in the defines around line 50

Bluetooth not working
After version ESP32Arduino_WordClockV060, NimBLE version 2.0 must be used.
For version ESP32Arduino_WordClockV060, only NimBLE version 1.4.3 works




If all else fails, try the 'Alternative software installation' method described below

## Alternative software installation

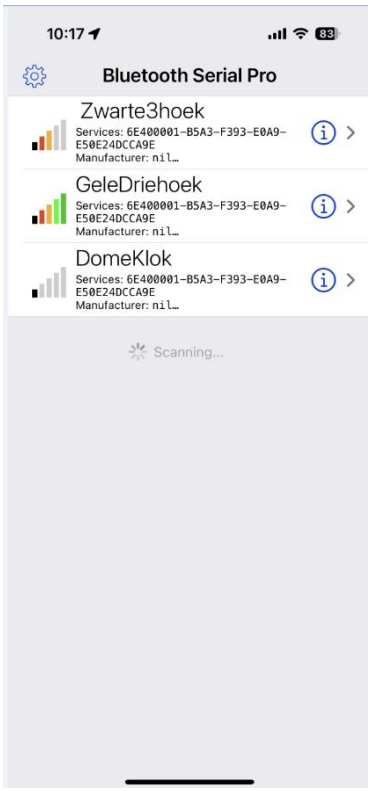



As described below, software can also be put 'Over the Air' on the Nano ESP32. The advantage is that you don't have to install the libraries and in a few years you will have problems with incompatible libraries.

The compilation is already done and the file uploaded to the Nano ESP32 is then saved as .bin file. Various .bin-file versions can be found on GitHub.
https://github.com/ednieuw/Arduino-ESP32-Nano-Wordclock

- Select the Nano ESP32 as the board (Tools->Board-ArduinoESPboards->Arduino Nano ESP32)
- In Examples, search for ArduinoOTA > and open the OTAWebUpdater program
- Choose Sketch->Upload or press the upload button in the top left corner
- Open the serial monitor and see which IP address is being printed.
  (Press the white reset button on the Nano ESP32 if nothing is printed)
- Type this IP address into a browser's URL (e.g.: 192.168.0.123)
- Login with admin and with password admin
- Find the .bin file and press update
  (For example 'ESP32_WordClockV100.inoNL114.bin' This is version V100 with settings for the NL114 clock)

## Clock operation



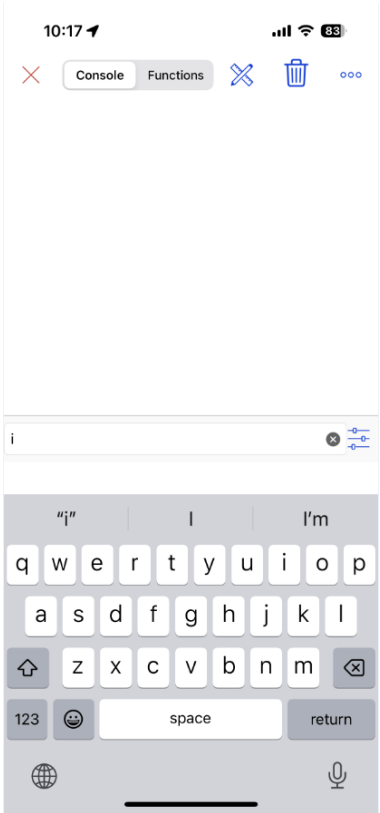

To connect to a WIFI network, an SSID (WIFI name) and password must be entered.
There are a few methods:
Connect the MCU in the clock to a PC with a micro-USB serial cable and use a serial terminal.
Use a BLE serial terminal app on a phone or tablet for connection.

For a PC, the Arduino IDE is recommended.
For IOS use : **BLE Serial Pro** or **BLESerial nRF.**
For Android use: **Serial Bluetooth terminal**.

Bluetooth Low Energy (BLE) can use two types of protocol: CC25nn or nRF52nn, where nn is a specific number. This clock uses nRF52.

- Launch the app and start a connection to the clock.

Some apps automatically launch with a connection window, but some require a connection icon to be pressed. There is probably one station, the word clock, to select from.




- Select the clock from the list.

- The app will show a window and a line where commands can be entered and sent to the clock.

- Sending the letter I or i for information will display the menu, followed by the actual settings of different preferences.

Inside the clock is an LED that has a red dot that lights up when the program is running. A green dot will light up when there is a WIFI connection. If there is a Bluetooth connection, a blue dot will illuminate in the LED.
In both cases, **you send the letter I of Information and the menu appears** .

Enter the first letter of the setting you want to change, followed by a code.
Some items only toggle On and Off. Like the W to turn WIFI Off or On.
To change the SSID and password: Send the letter **A** or **a** followed by

the name of the Wi-Fi station.

**Amy-SSID** and send this command. For example, AFRITZ! Box01 or aFRITZ! Box01. It doesn't matter if you don't have an uppercase or lowercase letter.

This is followed by the letter B with the password.
**Bmypassword** and send the password.

**Cbroadcastname** will change to the name displayed in the Bluetooth connection list. For example, something like: cWordClock

If the length of the SSID and/or password is less than 5 characters, the WIFI will be automatically disabled.
Use a minimum of 8 characters long for the SSID and password.
Check in the menu (third line from the bottom) if WIFI and NTP are enabled.

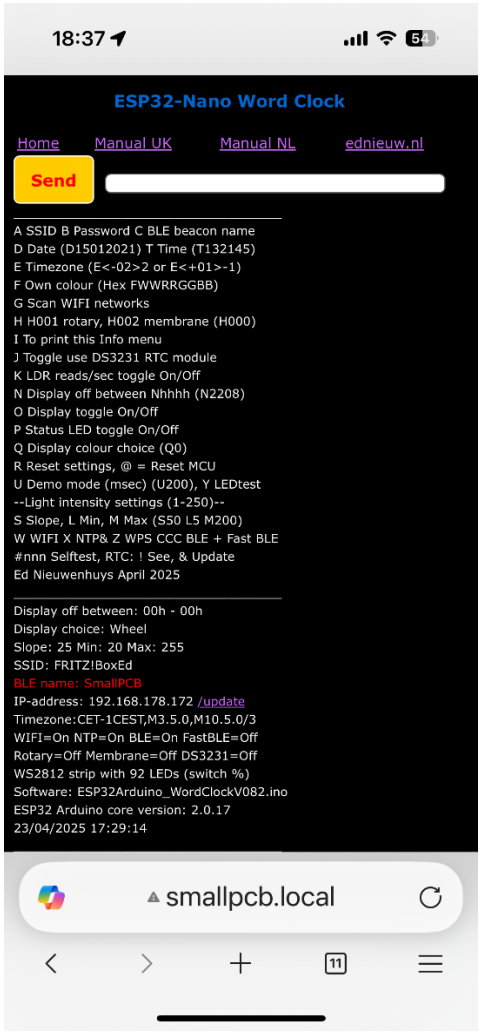
Enter @ to restart the Nano ESP32.
It will restart and connections will be made.
Sometimes the clock needs to be reset a second or third time before it can connect to WIFI.
If connection still does not connect, check the password you entered.
When WIFI is connected, the LED on the Nano ESP32 will flash green.

```
_____________________________
A SSID B Password C BLE beacon name
D Date (D15012021) T Time (T132145)
E Timezone (E<-02>2 or E<+01>-1)
F Own colour (Hex FWWRRGGBB)
G Scan WIFI networks
H H001 rotary, H002 membrane (H000)
I To print this Info menu
J Toggle use DS3231 RTC module
K LDR reads/sec toggle On/Off
N Display off between Nhhhh (N2208)
O Display toggle On/Off
P Status LED toggle On/Off
Q Display colour choice
  Q0 Yellow Q1 Hourly Q2 White
  Q3 All Own Q4 Own Q5 Wheel Q6 Dig
R Reset settings @ = Reset MCU
U Demo mode (msec) (U200)
--Light intensity settings (1-250)--
S Slope, L Min, M Max (S50 L5 M200)
W WIFI, X NTP&, CCC BLE, + Fast BLE
# Self test, ! See RTC, & Update RTC
Ed Nieuwenhuys December 2024
_____________________________
Display off between: 00h - 00h
Display choice: All Own
Slope: 50 Min: 5 Max: 255
SSID: FRITZ! Boxed
BLE name: SteelClock-01
IP address: 192.168.178.40 /update
Timezone:CET-1CEST,M3.5.0,M10.5.0/3
WIFI=On NTP=On BLE=On FastBLE=Off
Rotary=Off Membrane=Off DS3231=Off
SK6812 strip with 144 LEDs (switch %)
Software: ESP32Arduino_WordClockV062.ino
ESP32 Arduino core version: 2.0.17
30/12/2024 15:54:14
_____________________________________
```

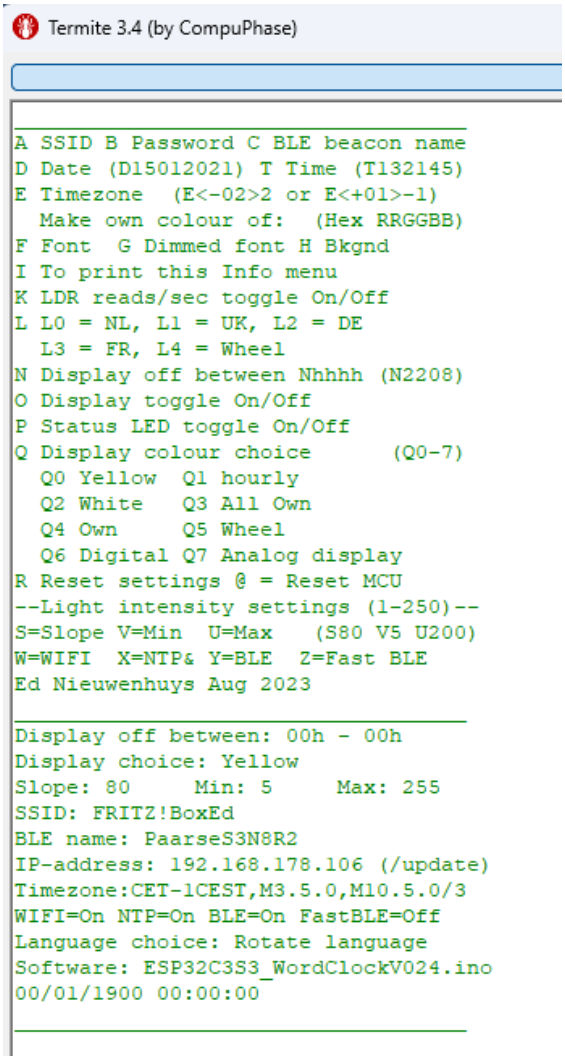
Menu displayed in serial output.

By default, the time zone is set to Amsterdam time.
A reset with option R in the menu will reset the time zone back to Amsterdam.
To set a different time zone, send the time zone string, preceded by the E or e sign.
Many time zones are printed at the bottom of this guide.
For example: if you live in Australia/Sydney, send the string, eAEST-10AEDT,M10.1.0,M4.1.0/3.
The clock uses daylight saving time (DST) when connected to an NTP server, but not when the DS3231 time module is used

ESP32Arduino_WoordKlokV100




HTML page on iPhone




'Termite' screen from a PC

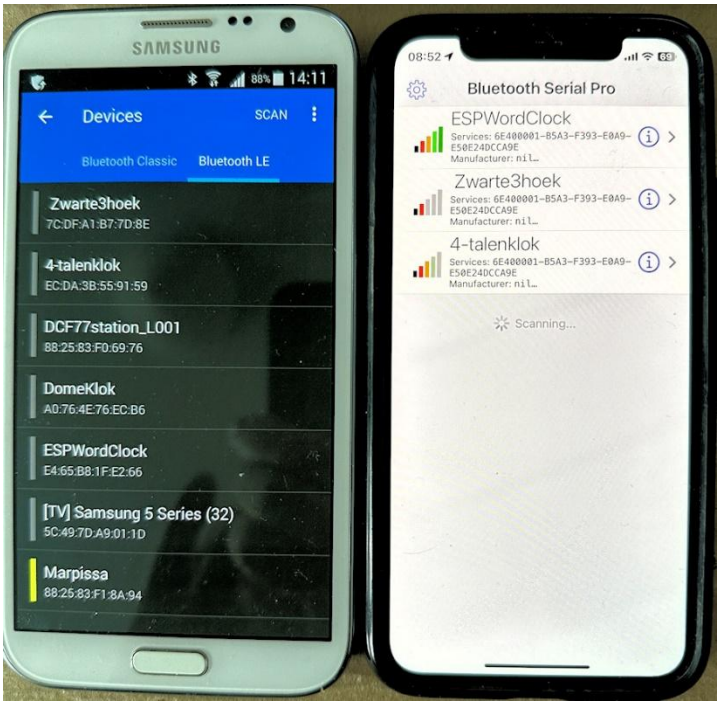
**Clock operation and settings**
The clock can be operated with the WIFI webpage or BLE UART terminal app.
When the clock is connected to WIFI, it has received an IP address from the router it is connected to. The IP address is printed in the menu.

To start the menu as a web page, the IP address of the clock must be entered into a web browser (for example: 192.168.178.77)
or use the BLE drive name followed by .local as the URL in the browser instead.
(Default after a reset: wordclock.local)

Or use a Bluetooth connection:
- Open the BLE terminal app.
- In the app, search for the clock to connect to and connect.

Each app has its own way of displaying the Bluetooth device to connect to.

The iPhone/iPad/iMac BLE serial apps are made by me and work with their default settings.
Other apps may need to change their sending and receiving data settings. Play around with the font size and the CR and LF setting until you
can speed up the transmission speed on an iPhone, iPad, or iMac with the BLE serial app by selecting the '+ Fast BLE' option from the menu
Unfortunately, some apps can't read strings longer than 20 characters, and you'll see the strings truncated or distorted.
If lines are printed incomplete, send in '+' to select a slower transmission mode.
If the transmission is too distorted and it is impossible to send the + sign, try the clock's web page and send a + sign.




If all else fails, you will need to connect the Arduino Nano ESP32 in the clock with a USB-C cable to a PC and send a + with the Arduino IDE or as a last resort R to factory reset.
(https://www.arduino.cc/en/software)

## Adjusting the light intensity of the display

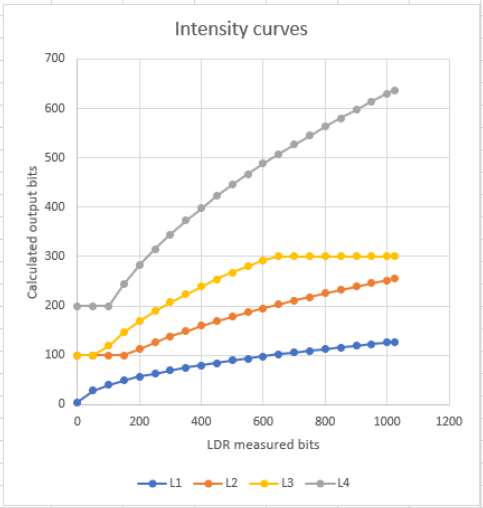
In the menu, the light intensity of the display can be controlled with three parameters:
--Light Intensity Settings (1-250)--
S=Slope L=Min M=Max (S80 L5 M200)
Default values are displayed between ().

S Quickly show the brightness reaches the maximum brightness.
I How bright the screen is in complete darkness.
M the maximum brightness of the screen .
The bottom half of the menu displays the stored values
Slope: 50 Min: 5 Max: 255

The clock reacts to light with its LDR (light-dependent resistance).
When it gets dark, the display does not switch off completely but remains dimmed at a minimum value.
The parameter L allows the lowest brightness to be controlled. With a value between 0 and 255, this brightness can be adjusted.
L5 is the default value.
The maximum brightness of the display is controlled by the parameter M. Also a value between 0 and 255.
Parameter S can be used to control the slope of how quickly the maximum brightness is reached.

| | | | | | |
|---|---|---|---|---|---|
| L | 5 | 100 | 100 | 200 | Lower |
| M | 200 | 400 | 300 | 999 | Upper |
| Y | 50 | 100 | 150 | 250 | Slope |

| LDR | L1 | L2 | L3 | L4 |
|---|---|---|---|---|
| 0 | 5 | 100 | 100 | 200 |
| 50 | 28 | 100 | 100 | 200 |
| 100 | 40 | 100 | 120 | 200 |
| 150 | 49 | 100 | 146 | 244 |
| 200 | 56 | 113 | 169 | 282 |
| 250 | 63 | 126 | 189 | 315 |
| 300 | 69 | 138 | 207 | 345 |
| 350 | 75 | 149 | 224 | 373 |
| 400 | 80 | 159 | 239 | 398 |
| 450 | 85 | 169 | 254 | 423 |
| 500 | 89 | 178 | 267 | 445 |
| 550 | 93 | 187 | 280 | 467 |
| 600 | 98 | 195 | 293 | 488 |
| 650 | 102 | 203 | 300 | 508 |
| 700 | 105 | 211 | 300 | 527 |
| 750 | 109 | 218 | 300 | 546 |
| 800 | 113 | 225 | 300 | 563 |
| 850 | 116 | 232 | 300 | 581 |
| 900 | 120 | 239 | 300 | 598 |
| 950 | 123 | 246 | 300 | 614 |
| 1000 | 126 | 252 | 300 | 630 |
| 1023 | 127 | 255 | 300 | 637 |

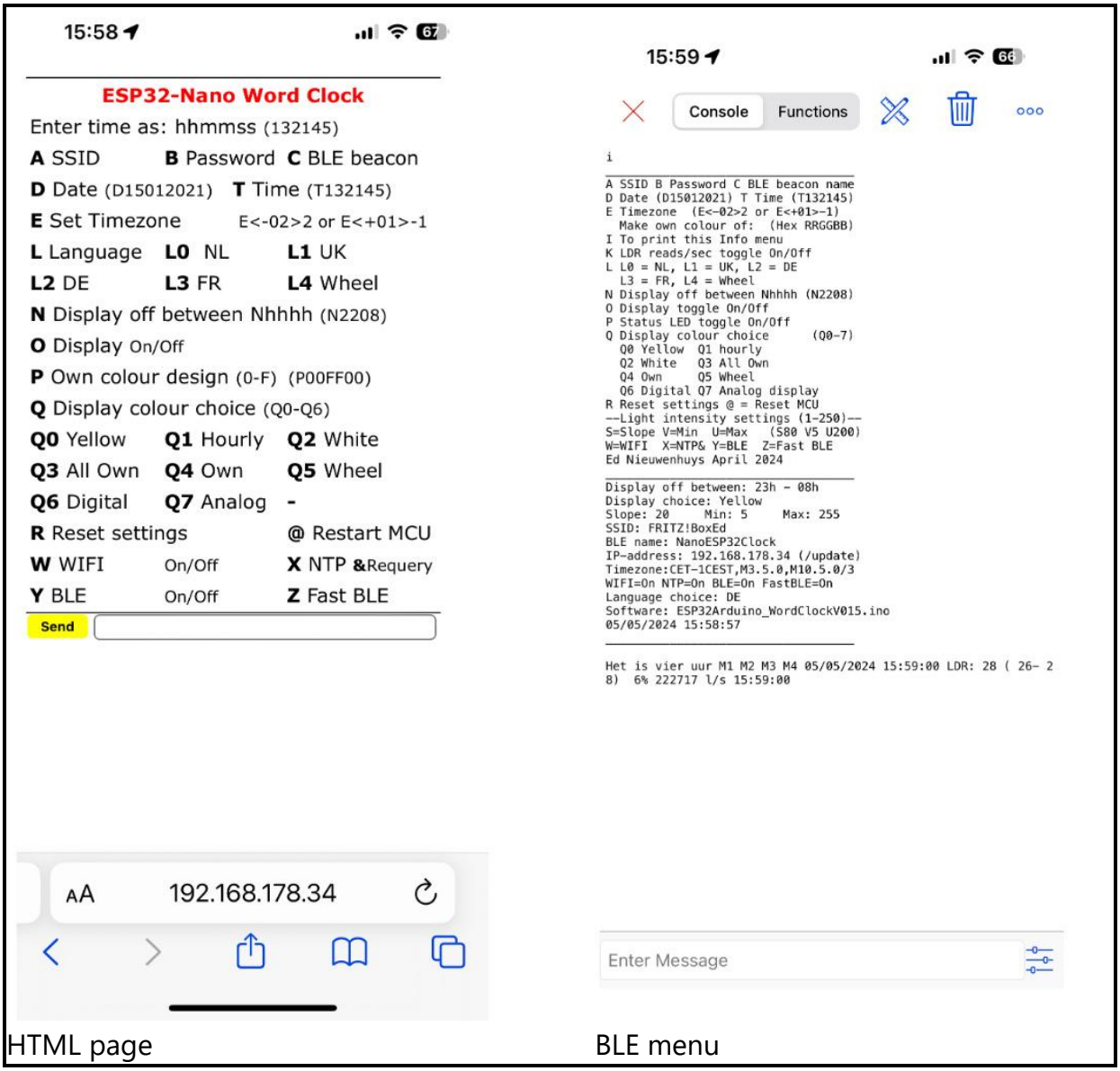


**Settings are set by entering the first character of a command, followed by parameters as needed.**

For example:
To set the colour of the words in the clock to white, enter: Q2

to turn of the clock LEDs between 22:00 and 8:00 send: n2208

Turn off WIFI by sending a W.
Restart the clock by sending the letter @.
Reset to default setting by sending R.

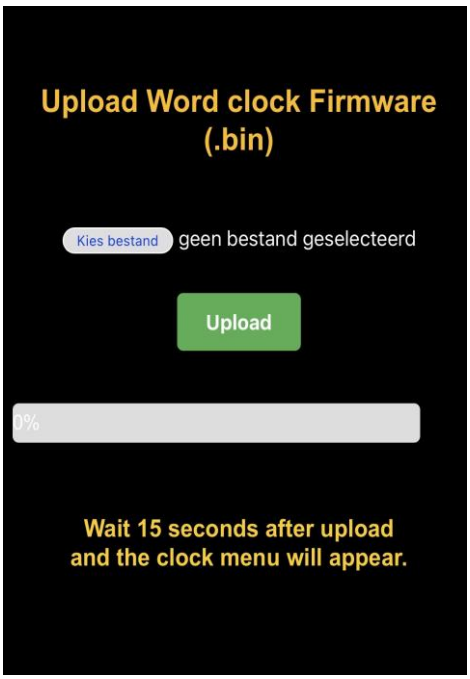


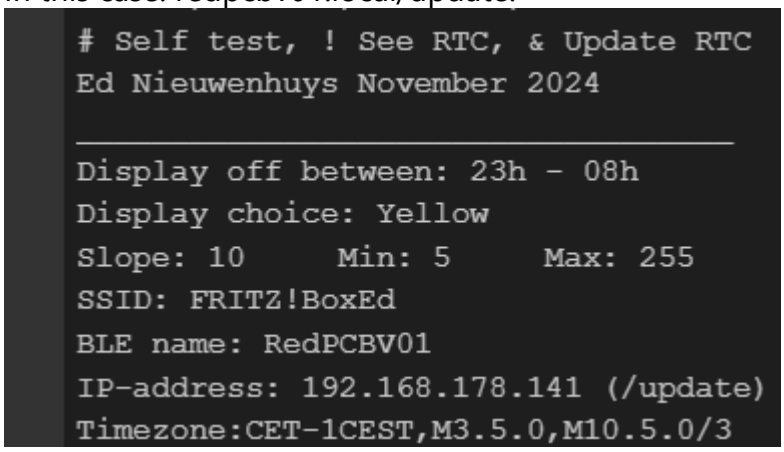

HTML page BLE menu

## Updating the software

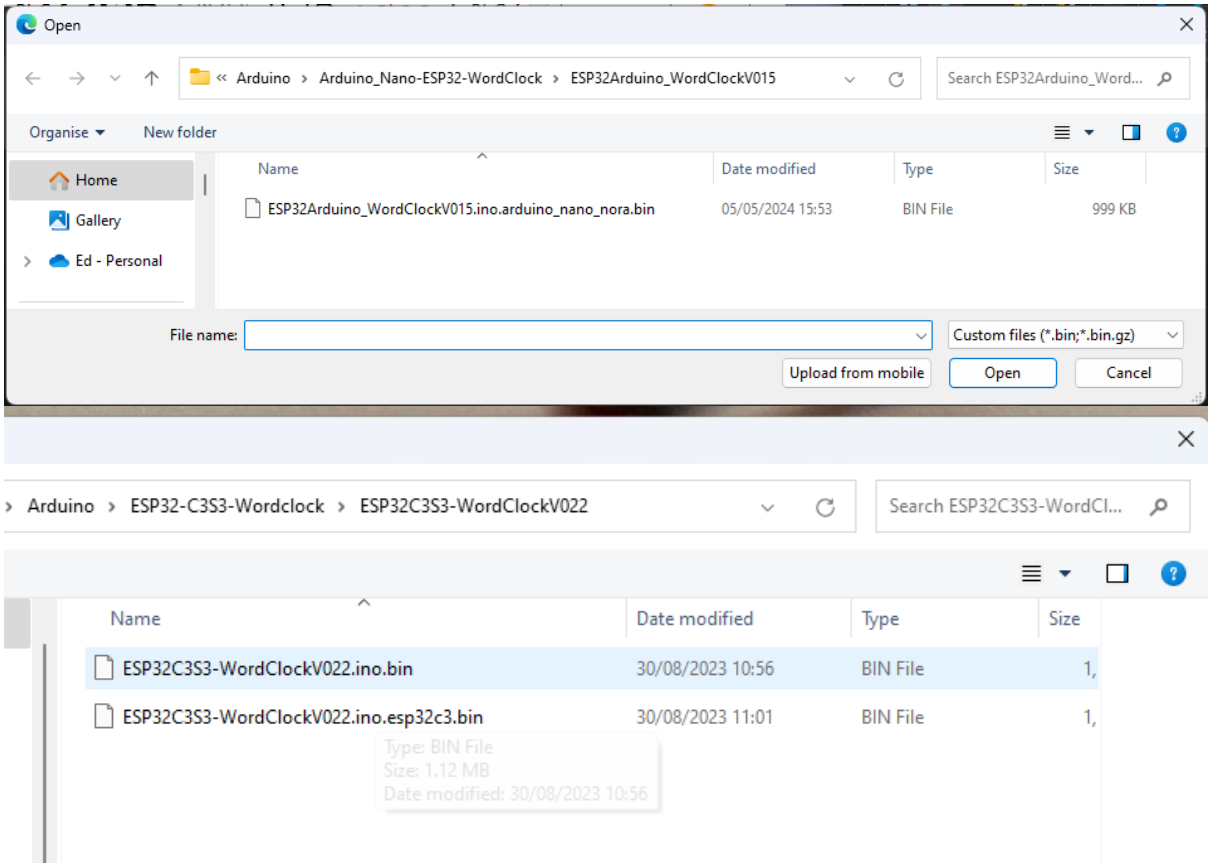
The software can be updated 'Over The Air' when the clock is connected to WIFI. The IP address is listed in the menu.

Enter the IP address of the clock, directly followed, without a space, by /update.
Something like this: 192.168.178.141/update.
Or use BLEbroadcastname.local/update instead of the IP address.
In this case: redpcbv01.local/update.




Choose File from the menu, and then select the bin file you want to update.
Something like: ESP32Arduino_WordClockV100.ino.bin
where V100 is the version number of the software.



## Detailed menu description

Many preferences can be set with the menu. These preferences are stored in the ESP32 storage space and are retained in the event of a power failure.

Enter the first character in the menu of the item to be changed, followed by the parameter. For example, N2208 to turn off the clock lights between 22:00 and 8:00. There is no difference between uppercase or lowercase letters. Both are OK.

When the clock is started and connected with a USC-C serial cable to a PC then in the 'serial terminal' of the Arduino IDE the boot log data are printed.

```
Serial started
Rotary NOT used
Rotary available
Found I2C address: 0X57
Found I2C address: 0X68
External RTC module IS found
DS3231 RTC software started
Mem.Checksum = 25065
Stored settings loaded
LED strip is SK6812
LED strip started
BLE started
10: networks found
 1: FRITZ! BoxEd -47 * -- Will connect to
 2: NETGEAR_EXT -69
 3: FRITZ! BoxEd -71 * -- Will connect to
 4: H369A209CE1 -75*
 5: H3baas -81*
 6: H369A209CE1 -81*
 7: FRITZ! Box -82 *
 8: H369A209CE1 -83*
 9: FRITZ! Box 5490 IS -83 *
10: FRITZ! Box 5490 XL -84 *
[WiFi-event] event: 0 : WiFi interface ready
[WiFi-event] event: 2 : WiFi client started
[WiFi-event] event: 4 : Connected to access point
[WiFi-event] event: 7 : Obtained IP address: 192.168.178.199
IP Address: 192.168.178.199
Web page started
WIFI started
01/01/2000 00:06:42
_____________________________________
_____________________________________
A SSID B Password C BLE beacon name
D Date (D15012021) T Time (T132145)
E Timezone (E<-02>2 or E<+01>-1)
F Own colour (Hex FWWRRGGBB)
G Scan WIFI networks
H H001 rotary, H002 membrane (H000)
I Info menu, II long menu
J Toggle use DS3231 RTC module
K LDR reads/sec toggle On/Off
N Display off between Nhhhh (N2208)
O Display toggle On/Off
P Status LED toggle On/Off
Q Display colour choice (Q0)
R Reset settings, @ = Reset MCU
U Demo (msec) (U200) Y LED test
--Light intensity settings (1-250)--
S Slope, L Min, M Max (S50 L5 M200)
W WIFI X NTP& Z WPS CCC BLE + Fast BLE
#nnn Selftest, RTC: ! See, & Update
Ed Nieuwenhuys April 2025
_____________________________________
```

**A SSID B Password C BLE beacon name**

Enter the name of the SSID and password of the router to which the clock is to be connected. The Bluetooth broadcast name displayed in the Bluetooth serial terminal app can be changed.
For example: **aFRITZ! Box** or **AFRITZ! Box** .
Then enter the password. For example: **Password**.
If necessary, enter **cBluetoothName** (default after a reset: wordclock )
Restart the clock by sending @ .

**CCC** Entering CCC enables or disables BLE.
Be careful when turning it off. When BLE is off, the clock can only be controlled with WIFI or the USB serial port.

**D Set Date** and **T Set Time**

If there is no connection to WIFI time and an RTC DS3231 is connected, the date must be set manually.
For example, enter D06112022 to set the date to November 6, 2022.
For example, enter T132145 (or 132145, or t132145) to set the time to 45 seconds and 21 minutes after one hour.
Changing the date and time only works if WIFI and NTP are disabled.

**E Set time zone E<-02>2 or E<+01>-1**

At the bottom of this page are the time zones as used in 2022.
It is quite a complicated string and it is therefore wise to copy it.
Let's choose one if you live here: Antarctica/Troll,"<+00>0<+02>-2,M3.5.0/1,M10.5.0/3"
Copy the string between the " "'s and send it starting with an 'E' or 'e' in front of it.
E<+00>0<+02>-2,M3.5.0/1,M10.5.0/3

**F Custom color (Hex WWRRGGBB)**

The colour of the words displayed in the clock can be adjusted and can be selected with option Q3 or Q4 in the menu.
The format to be entered is hexadecimal. 0123456789ABCDEF are the characters that can be used. The command consists of 2 digits for white followed by 2 digits for Red followed by 2 digits for Green and ending with 2 digits for Blue.
To colour the letters intense Green, enter **f0000FF00**, preceded by the letter F.
To colour the letters intense Red, enter 00FF0000, preceded by the character F.
To set intense blue, enter F000000FF.
WS2812 LED strips do not have a white LED, and the command to control the white LEDs has no effect.
With SK6812 LEDs, an additional white LED is used in addition to the three RGB LEDs in the same housing.
For example: F8800FF00 is 50% white and 100% green.
In short: 00 is LED off, 44 is 25%, 88 is 50%, BB is 75% and FF is 100%.

**G Scan for nearby WIFI networks**

Sending 'G' will print the available nearby networks in the serial terminal.

**H H001 Rotary, H002 Membrane (H000).**

By sending 'Hnnn' you can choose whether an optional dial or membrane button is built into the clock. The software then checks whether the buttons are pressed.
After a Reset to factory setting, the use of a rotary or membrane button will be left unchanged.
If H001 or H002 is inserted and no rotary or diaphragm knob is connected, this may result in unwanted input.

**I or ii to show the info menu,**

Print the menu.
With the option I in the menu, you can choose between a short menu for everyday use or a long menu with option II (2x i) for an extensive menu with all options.

**J Switching to use the DS3231 module time**

Sending 'J' will turn the use of an optional DS3231 time module ON or OFF. If the clock doesn't have an internet connection, time will likely pass quickly. By using turning on the DS3231 time module, the deviation is reduced to a few seconds per year.
The time can be entered using options T and D in the menu.
If the time is not visible, send the command & to show the internal clock times

**K Reads/sec switches On/Off**

By entering a K, the printing of the LDR reading changes the measured light intensity. It also shows how many times per second the processor runs through the program and checks its tasks to keep the clock running.

```
TestLDR: On
LDR reading, %Out, loops per second and time
LDR: 1= 1% 205413 l/s 16:06:08
LDR: 1= 1% 215535 l/s 16:06:09
LDR: 1= 1% 215451 l/s 16:06:10
LDR: 1= 1% 215350 l/s 16:06:11
TestLDR: Off
```

**N Display off between Nhhhh (N2208)**

With N2208, the display will turn off between 22:00 and 8:00.

**O Switch display on/off**

O Turns the display off and on.

**P Switching Status LEDs Off and On**

P Turns the status LEDs on the Arduino Nano off and on.

**Q Choice of display colour (Q0-6)**
**Q0 Yellow Q1 per hour Q2 White Q3 All Custom Q4 Custom Q5 Wheel Q6 Digital display**

Q0 shows the time with yellow words
and HET (or IT) will change from green via yellow to red within an hour.
and will change IS or WAS from green to red via yellow in a minute.
Q1 will show a different colour every hour.
Q2 shows all texts in white.
Q3 and Q4 use their own defined colours. (With option F in the menu)
Q5 will go through the rainbow colours in an hour.
Q6 is a digital display
Q8 Toggles extra optional LEDs On/Off
Q9 Toggles HET/IS/WAS (IT/IS/WAS) On/Off

**R Reset settings**
R resets all preferences to the default settings.
The SSID, password, BLE name, and time zone aren't cleared. RRR knew these last four. WIFI, NTP and BLE are turned on
RRRRR is the combination of the option R and RRR. The type of LED strip and the use of a rotary or membrane button are left unchanged.
Perform a reset with RRRRR when the sketch is uploaded to the Arduino Nano ESP32 for the first time.

**S=Slope L=Min M=Max (S50 L5 M200)**
S How quickly the brightness reaches maximum brightness.
L How bright the display is in complete darkness.
M the maximum brightness of the display.
Values between 0 and 255.

**U Demo Mode (msec) (U200)**
Enter U, followed by the duration of one second in milliseconds.
U200 (200 milliseconds) speeds up the clock 5 times.
Sending only a U disables the demo mode.

**Y LED Test**
All LEDs will go through a rainbow.
The test stops by itself.

**Z Start WPS**
Start WPS in the clock by entering Z in the menu and start WPS on the WIFI router.
The SSID and password of the WIFI router are permanently stored in the clock's memory. In the clock menu, the SSID of the router will be visible.

**W=WIFI, X=NTP & , CCC=BLE**

Turn WIFI, NTP on and off.
Enter the character to enable or disable the option.
It is indicated at the bottom of the menu.

```
     Display off: 00h - 00h
  Display choice: Yellow
            SSID: FRITZ!BoxEd
        BLE name: ESPWordClock
      IP-address: 192.168.178.78
Timezone:CET-1CEST,M3.5.0,M10.5.0/3
 WIFI=On NTP=On BLE=On FastBLE=On
 Language choice: Rotate language
```

By sending a & , the NTP time is put into the internal Nano ESP32 RTC and DS3231 time module.

**+ Fast BLE**

The BLE UART protocol transmits standard packets that are 20 bytes long. There is a delay of 50 msec between each packet.
The IOS BLEserial app, and perhaps others as well, can receive packets of 80 bytes or more before characters are missed. This will speed up the menu printing.
Option + switches between the long and short packages.

**! = Show NTP, RTC and DS3231 time !** will display the NTP, RTC and DS3231 time as they are stored in the clock in the clock. The DS3231 time module must be installed to show a realistic time for the module. Same as the & option but this option will not be updated from the internet NTP server but only shows the time.

**#nnn = Self-test**

By sending a #, the clock self-test starts. This is useful for checking that all the words in the clock are functioning.
From V062 #nnnn is also possible. Nnnn are the milliseconds of delay between each word.
The self-test stops on its own after a cycle.

**% = Switching between SK6812 and WS2812 LED strip**

With this option, the used LED strip can be changed. The clock is equipped with one of these two types of LED strips.
A reset of all settings by sending an R in the menu does not change the selection of the LED strip.

**$ =Fireside**

Shows a fireside on and 144 LED clock**.** On other types of clocks, the result will be less beautiful.

**@ = Reset MCU**

@ restart the MCU. This is useful if the SSID, etc. are changed and the program needs to be restarted.
Settings are retained.

### & = Does the RTC module synchronise the NTP time

The RTC time module in the clock can control the time in the clock. The T and D menu options allow the time and date to be entered manually. This option copies the NTP time to the RTC module.

The computer that is in the clock has its own internal clock. It expires seconds per day and is updated regularly with or the NTP time to the DS3231 RTC time module. If NTP and the DS3231 RTC time module are switched off, the clock will display a time, but it will get out of sync with the correct time after a few days.

### 123456 Setting Time in DS3231 RTC Time Module

Enter the time as 152300 hhmmss. Same as T152300.

Changing date and time only works if NTP is disabled.

### Time zone

Copy the text **between the quotation marks** and paste it after the letter e or E and send it to the clock.

For the Netherlands and Belgium send: eCET-1CEST,M3.5.0,M10.5.0/3

```
Africa/Abidjan,'GMT0'
Africa/Accra,'GMT0'
Africa/Addis_Ababa,'EAT-3'
Africa/Algiers,'CET-1'
Africa/Asmara, "EAT-3"
Africa/Bamako,'GMT0'
Africa/Bangui,"WAT-1"
Africa/Banjul,'GMT0'
Africa/Bissau,'GMT0'
Africa/Blantyre,"CAT-2"
Africa/Brazzaville,"WAT-1"
Africa/Bujumbura,'CAT-2'
Africa/Cairo, "EET-2"
Africa/Casablanca,"<+01>-1"
Africa/Ceuta,"CET-1CEST,M3.5.0,M10.5.0/3"
Africa/Conakry,"GMT0"
Africa/Dakar,'GMT0'
Africa/Dar es Salaam,"EAT-3"
Africa/Djibouti,"EAT-3"
Africa/Douala,"WHAT-1"
Africa/El Aaiun,"<+01>-1"
Africa/Freetown,'GMT0'
Africa/Gaborone,"CAT-2"
Africa/Harare, "CAT-2"
Africa/Johannesburg,'SAST-2'
Africa/Juba,"CAT-2"
Africa/Kampala, "EAT-3"
Africa/Khartoum,"CAT-2"
Africa/Kigali,"CAT-2"
Africa/Kinshasa,"WHAT-1"
Africa/Lagos,"WAT-1"
Africa/Libreville,"WHAT-1"
Africa/Lome,'GMT0'
Africa/Luanda, "WHAT-1"
Africa/Lubumbashi,"CAT-2"
Africa/Lusaka,"CAT-2"
Africa/Malabo, "WHAT-1"
Africa/Maputo,"CAT-2"
Africa/Maseru,"SAST-2"
Africa/Mbabane,'SAST-2'
Africa/Mogadishu,'EAT-3'
Africa/Monrovia,'GMT0'
Africa/Nairobi,'EAT-3'
Africa/Ndjamena,"WAT-1"
Africa/Niamey,"WAT-1"
Africa/Nouakchott,'GMT0'
Africa/Ouagadougou,'GMT0'
Africa/Porto-Novo, "WAT-1"
Africa/Sao Tome,'GMT0'
Africa/Tripoli, "EET-2"
Africa/Tunis, "CET-1"
Africa/Windhoek,"CAT-2"
America/Adak,"HST10DT,M3.2.0,M11.1.0"
America/Anchorage,"AKST9AKDT,M3.2.0,M11.1.0"
America/Anguilla, "AST4"
America/Antigua, "AST4"
America/Araguaina,"<-03>3"
America/Argentina/Buenos_Aires,"<-03>3"
America/Argentina/Catamarca,'<-03>3'
America/Argentina/Cordoba,"<-03>3"
America/Argentina/Jujuy,"<-03>3"
America/Argentina/La_Rioja,"<-03>3"
America/Argentina/Mendoza,"<-03>3"
America/Argentina/Rio_Gallegos,"<-03>3"
America/Argentina/Salta,"<-03>3"
America/Argentina/San_Juan,"<-03>3"
America/Argentina/San_Luis,"<-03>3"
America/Argentina/Tucuman,"<-03>3"
America/Argentina/Ushuaia,"<-03>3"
America/Aruba, "AST4"
America/Asuncion,"<-04>4<-03>,M10.1.0/0,M3.4.0/0"
America/Atikokan,"EST5"
America/Bahia,"<-03>3"
America/Bahia_Banderas,"CST6CDT,M4.1.0,M10.5.0"
America/Barbados, "AST4"
America/Belem,"<-03>3"
America/Belize, "CST6"
America/Blanc-Sablon, "AST4"
America/Boa Vista,"<-04>4"
America/Bogota,"<-05>5"
America/Boise,"MST7MDT,M3.2.0,M11.1.0"
America/Cambridge_Bay,"MST7MDT,M3.2.0,M11.1.0"
America/Campo Grande,"<-04>4"
America/Cancun, "EST5"
America/Caracas,"<-04>4"
America/Cayenne,"<-03>3"
America/Cayman,"EST5"
America/Chicago,"CST6CDT,M3.2.0,M11.1.0"
America/Chihuahua,"MST7MDT,M4.1.0,M10.5.0"
America/Costa_Rica,"CST6"
America/Creston,"MST7"
America/Cuiaba,"<-04>4"
America/Curacao, "AST4"
America/Danmarkshavn,'GMT0'
America/Dawson,"MST7"
America/Dawson Creek,"MST7"
America/Denver,"MST7MDT,M3.2.0,M11.1.0"
America/Detroit,"EST5EDT,M3.2.0,M11.1.0"
America/Dominica, "AST4"
America/Edmonton,"MST7MDT,M3.2.0,M11.1.0"
America/Eirunepe,"<-05>5"
America/El_Salvador,"CST6"
America/Fortaleza,"<-03>3"
America/Fort_Nelson,"MST7"
America/Glace_Bay,"AST4ADT,M3.2.0,M11.1.0"
America/Godthab,"<-03>3<-02>,M3.5.0/-2,M10.5.0/-1"
America/Goose_Bay,"AST4ADT,M3.2.0,M11.1.0"
America/Grand_Turk,"EST5EDT,M3.2.0,M11.1.0"
America/Grenada, "AST4"
America/Guadeloupe, "AST4"
America/Guatemala, "CST6"
America/Guayaquil,"<-05>5"
America/Guyana,"<-04>4"
America/Halifax,"AST4ADT,M3.2.0,M11.1.0"
America/Havana,"CST5CDT,M3.2.0/0,M11.1.0/1"
America/Hermosillo, "MST7"
America/Indiana/Indianapolis,"EST5EDT,M3.2.0,M11.1.0"
America/Indiana/Knox,"CST6CDT,M3.2.0,M11.1.0"
America/Indiana/Marengo,"EST5EDT,M3.2.0,M11.1.0"
America/Indiana/Petersburg,"EST5EDT,M3.2.0,M11.1.0"
America/Indiana/Tell_City,"CST6CDT,M3.2.0,M11.1.0"
America/Indiana/Vevay,"EST5EDT,M3.2.0,M11.1.0"
America/Indiana/Vincennes,"EST5EDT,M3.2.0,M11.1.0"
America/Indiana/Winamac,"EST5EDT,M3.2.0,M11.1.0"
America/Inuvik,"MST7MDT,M3.2.0,M11.1.0"
America/Iqaluit,"EST5EDT,M3.2.0,M11.1.0"
America/Jamaica,"EST5"
America/Juneau,"AKST9AKDT,M3.2.0,M11.1.0"
America/Kentucky/Louisville,"EST5EDT,M3.2.0,M11.1.0"
America/Kentucky/Monticello,"EST5EDT,M3.2.0,M11.1.0"
America/Kralendijk, "AST4"
America/La_Paz,"<-04>4"
America/Lima,"<-05>5"
America/Los_Angeles,"PST8PDT,M3.2.0,M11.1.0"
America/Lower_Princes,"AST4"
America/Maceio,"<-03>3"
America/Managua, "CST6"
America/Manaus,"<-04>4"
America/Marigot, "AST4"
America/Martinique, "AST4"
America/Matamoros,"CST6CDT,M3.2.0,M11.1.0"
America/Mazatlan,"MST7MDT,M4.1.0,M10.5.0"
America/Menominee,"CST6CDT,M3.2.0,M11.1.0"
America/Merida,"CST6CDT,M4.1.0,M10.5.0"
America/Metlakatla,"AKST9AKDT,M3.2.0,M11.1.0"
America/Mexico_City,"CST6CDT,M4.1.0,M10.5.0"
America/Miquelon,"<-03>3<-02>,M3.2.0,M11.1.0"
America/Moncton,"AST4ADT,M3.2.0,M11.1.0"
America/Monterrey,"CST6CDT,M4.1.0,M10.5.0"
America/Montevideo,"<-03>3"
America/Montreal,"EST5EDT,M3.2.0,M11.1.0"
America/Montserrat,"AST4"
America/Nassau,"EST5EDT,M3.2.0,M11.1.0"
America/New_York,"EST5EDT,M3.2.0,M11.1.0"
America/Nipigon,"EST5EDT,M3.2.0,M11.1.0"
America/Nome,"AKST9AKDT,M3.2.0,M11.1.0"
America/Noronha,"<-02>2"
America/North_Dakota/Beulah,"CST6CDT,M3.2.0,M11.1.0"
America/North_Dakota/Center,"CST6CDT,M3.2.0,M11.1.0"
America/North_Dakota/New_Salem,"CST6CDT,M3.2.0,M11.1.0"
America/Nuuk,"<-03>3<-02>,M3.5.0/-2,M10.5.0/-1"
America/Ojinaga,"MST7MDT,M3.2.0,M11.1.0"
America/Panama, "EST5"
America/Pangnirtung,"EST5EDT,M3.2.0,M11.1.0"
America/Paramaribo,"<-03>3"
America/Phoenix, "MST7"
America/Port-au-Prince,"EST5EDT,M3.2.0,M11.1.0"
America/Port_of_Spain,"AST4"
America/Porto_Velho,"<-04>4"
America/Puerto_Rico,"AST4"
America/Punta_Arenas,"<-03>3"
America/Rainy_River,"CST6CDT,M3.2.0,M11.1.0"
America/Rankin_Inlet,"CST6CDT,M3.2.0,M11.1.0"
America/Recife,"<-03>3"
America/Regina, "CST6"
America/Resolute,"CST6CDT,M3.2.0,M11.1.0"
America/Rio_Branco,"<-05>5"
America/Santarem,"<-03>3"
America/Santiago,"<-04>4<-03>,M9.1.6/24,M4.1.6/24"
America/Santo_Domingo,"AST4"
America/Sao_Paulo,"<-03>3"
America/Scoresbysund,"<-01>1<+00>,M3.5.0/0,M10.5.0/1"
America/Sitka,"AKST9AKDT,M3.2.0,M11.1.0"
America/St_Barthelemy,"AST4"
America/St_Johns,"NST3:30NDT,M3.2.0,M11.1.0"
America/St_Kitts,"AST4"
America/St_Lucia,"AST4"
America/St_Thomas,"AST4"
America/St_Vincent,"AST4"
America/Swift_Current,"CST6"
America/Tegucigalpa, "CST6"
America/Thule,"AST4ADT,M3.2.0,M11.1.0"
America/Thunder_Bay,"EST5EDT,M3.2.0,M11.1.0"
America/Tijuana,"PST8PDT,M3.2.0,M11.1.0"
America/Toronto,"EST5EDT,M3.2.0,M11.1.0"
America/Tortola, "AST4"
America/Vancouver,"PST8PDT,M3.2.0,M11.1.0"
America/Whitehorse,"MST7"
America/Winnipeg,"CST6CDT,M3.2.0,M11.1.0"
America/Yakutat,"AKST9AKDT,M3.2.0,M11.1.0"
America/Yellowknife,"MST7MDT,M3.2.0,M11.1.0"
Antarctica/Casey,"<+11>-11"
Antarctica/Davis,"<+07>-7"
Antarctica/DumontDUrville,"<+10>-10"
Antarctica/Macquarie,"AEST-
10ADET,M10.1.0,M4.1.0/3"
Antarctica/Mawson,"<+05>-5"
Antarctica/McMurdo,"NZST-12NZDT,M9.5.0,M4.1.0/3"
Antarctica/Palmer,"<-03>3"
Antarctica/Rothera,"<-03>3"
Antarctica/Syowa,"<+03>-3"
Antarctica/Troll,"<+00>0<+02>-
2,M3.5.0/1,M10.5.0/3"
Antarctica/Vostok,"<+06>-6"
Arctic/Longyearbyen,"CET-1CEST,M3.5.0,M10.5.0/3"
Asia/Aden,"<+03>-3"
Asia/Almaty,"<+06>-6"
Asia/Amman,"EET-2EEST,M2.5.4/24,M10.5.5/1"
Asia/Anadyr,"<+12>-12"
Asia/Aqtau,"<+05>-5"
Asia/Aqtobe,"<+05>-5"
Asia/Ashgabat,"<+05>-5"
Asia/Atyrau,"<+05>-5"
Asia/Baghdad,"<+03>-3"
Asia/Bahrain,"<+03>-3"
Asia/Baku,"<+04>-4"
Asia/Bangkok,"<+07>-7"
Asia/Barnaul,"<+07>-7"
Asia/Beirut,"EET-2EEST,M3.5.0/0,M10.5.0/0"
Asia/Bishkek,"<+06>-6"
Asia/Brunei,"<+08>-8"
Asia/Chita,"<+09>-9"
Asia/Choibalsan,"<+08>-8"
Asia/Colombo,"<+0530>-5:30"
Asia/Damascus,"EET-2EEST,M3.5.5/0,M10.5.5/0"
Asia/Dhaka,"<+06>-6"
Asia/Dili,"<+09>-9"
Asia/Dubai,"<+04>-4"
Asia/Dushanbe,"<+05>-5"
Asia/Famagusta,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Asia/Gaza,"EET-2EEST,M3.4.4/48,M10.5.5/1"
Asia/Hebron,"EET-2EEST,M3.4.4/48,M10.5.5/1"
Asia/Ho_Chi_Minh,"<+07>-7"
Asia/Hong_Kong,"HKT-8"
Asia/Hovd,"<+07>-7"
Asia/Irkutsk,"<+08>-8"
Asia/Jakarta,"WIB-7"
Asia/Jayapura,"WHITE-9"
Asia/Jerusalem,"IST-2IDT,M3.4.4/26,M10.5.0"
Asia/Kabul,"<+0430>-4:30"
Asia/Kamchatka,"<+12>-12"
Asia/Karachi, "PKT-5"
Asia/Kathmandu,"<+0545>-5:45"
Asia/Khandyga,"<+09>-9"
Asia/Kolkata,"IST-5:30"
Asia/Krasnoyarsk,"<+07>-7"
Asia/Kuala_Lumpur,"<+08>-8"
Asia/Kuching,"<+08>-8"
Asia/Kuwait,"<+03>-3"
Asia/Macau, "CST-8"
Asia/Magadan,"<+11>-11"
Asia/Makassar,"WITA-8"
Asia/Manila, "PST-8"
Asia/Muscat,"<+04>-4"
Asia/Nicosia,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Asia/Novokuznetsk,"<+07>-7"
Asia/Novosibirsk,"<+07>-7"
Asia/Omsk,"<+06>-6"
Asia/Oral,"<+05>-5"
Asia/Phnom_Penh,"<+07>-7"
Asia/Pontianak, "WIB-7"
Asia/Pyongyang,"KST-9"
Asia/Qatar,"<+03>-3"
Asia/Qyzylorda,"<+05>-5"
Asia/Riyadh,"<+03>-3"
Asia/Sakhalin,"<+11>-11"
Asia/Samarkand,"<+05>-5"
Asia/Seoul,"KST-9"
Asia/Shanghai, "CST-8"
Asia/Singapore,"<+08>-8"
Asia/Srednekolymsk,"<+11>-11"
Asia/Taipei,"CST-8"
Asia/Tashkent,"<+05>-5"
Asia/Tbilisi,"<+04>-4"
Asia/Tehran,"<+0330>-3:30<+0430>,J79/24,J263/24"
Asia/Thimphu,"<+06>-6"
Asia/Tokyo,"JST-9"
Asia/Tomsk,"<+07>-7"
Asia/Ulaanbaatar,"<+08>-8"
Asia/Urumqi,"<+06>-6"
Asia/Ust-Nera,"<+10>-10"
Asia/Vientiane,"<+07>-7"
Asia/Vladivostok,"<+10>-10"
Asia/Yakutsk,"<+09>-9"
Asia/Yangon,"<+0630>-6:30"
Asia/Yekaterinburg,"<+05>-5"
Asia/Yerevan,"<+04>-4"
Atlantic/Azores,"<-01>1<+00>,M3.5.0/0,M10.5.0/1"
Atlantic/Bermuda,"AST4ADT,M3.2.0,M11.1.0"
Atlantic/Canary,"WET0WEST,M3.5.0/1,M10.5.0"
Atlantic/Cape_Verde,"<-01>1"
Atlantic/Faroe,"WET0WEST,M3.5.0/1,M10.5.0"
Atlantic/Madeira,"WET0WEST,M3.5.0/1,M10.5.0"
Atlantic/Reykjavik,'GMT0'
Atlantic/South_Georgia,"<-02>2"
Atlantic/Stanley,"<-03>3"
Atlantic/St_Helena,'GMT0'
Australia/Adelaide,"ACST-9:30ACDT,M10.1.0,M4.1.0/3"
Australia/Brisbane,'AEST-10'
Australia/Broken_Hill,"ACST-
9:30ACDT,M10.1.0,M4.1.0/3"
Australia/Currie,"AEST-10AEDT,M10.1.0,M4.1.0/3"
Australia/Darwin,"ACST-9:30"
Australia/Eucla,"<+0845>-8:45"
Australia/Hobart,"AEST-10AEDT,M10.1.0,M4.1.0/3"
Australia/Lindeman,"AEST-10"
Australia/Lord_Howe,"<+1030>-10:30<+11>-
11,M10.1.0,M4.1.0"
Australia/Melbourne,"AEST-10AEDT,M10.1.0,M4.1.0/3"
Australia/Perth, "AWST-8"
Australia/Sydney,"AEST-10AEDT,M10.1.0,M4.1.0/3"
Europe/Amsterdam,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Andorra,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Astrakhan,"<+04>-4"
Europe/Athens,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Belgrade,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Berlin,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Bratislava,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Brussels,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Bucharest,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Budapest,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Busingen,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Chisinau,"EET-2EEST,M3.5.0,M10.5.0/3"
Europe/Copenhagen,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Dublin,"IST-1GMT0,M10.5.0,M3.5.0/1"
Europe/Gibraltar,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Guernsey,"GMT0BST,M3.5.0/1,M10.5.0"
Europe/Helsinki,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Isle_of_Man,"GMT0BST,M3.5.0/1,M10.5.0"
Europe/Istanbul,"<+03>-3"
Europe/Jersey,"GMT0BST,M3.5.0/1,M10.5.0"
Europe/Kaliningrad,"EET-2"
Europe/Kiev,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Kirov,"<+03>-3"
Europe/Lisbon,"WET0WEST,M3.5.0/1,M10.5.0"
Europe/Ljubljana,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/London,"GMT0BST,M3.5.0/1,M10.5.0"
Europe/Luxembourg,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Madrid,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Malta,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Mariehamn,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Minsk,"<+03>-3"
Europe/Monaco,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Moscow,"MSK-3"
Europe/Oslo,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Paris,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Podgorica,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Prague,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Riga,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Rome,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Samara,"<+04>-4"
Europe/San_Marino,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Sarajevo,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Saratov,"<+04>-4"
Europe/Simferopol,"MSK-3"
Europe/Skopje,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Sofia,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Stockholm,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Tallinn,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Tirane,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Ulyanovsk,"<+04>-4"
Europe/Uzhgorod,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Vaduz,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Vatican,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Vienna,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Vilnius,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Volgograd,"<+03>-3"
Europe/Warsaw,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Zagreb,"CET-1CEST,M3.5.0,M10.5.0/3"
Europe/Zaporozhye,"EET-2EEST,M3.5.0/3,M10.5.0/4"
Europe/Zurich,"CET-1CEST,M3.5.0,M10.5.0/3"
Indian/Antananarivo, "EAT-3"
Indian/Chagos,"<+06>-6"
Indian/Christmas,"<+07>-7"
Indian/Cocos,"<+0630>-6:30"
Indian/Comoro, "EAT-3"
Indian/Kerguelen,"<+05>-5"
Indian/Mahe,"<+04>-4"
Indian/Maldives,"<+05>-5"
Indian/Mauritius,"<+04>-4"
Indian/Mayotte,"EAT-3"
Indian/Reunion,"<+04>-4"
Pacific/Apia,"<+13>-13"
Pacific/Auckland,"NZST-
12NZDT,M9.5.0,M4.1.0/3"
Pacific/Bougainvillea,"<+11>-11"
Pacific/Chatham,"<+1245>-
12:45<+1345>,M9.5.0/2:45,M4.1.0/3:45"
Pacific/Chuuk,"<+10>-10"
Pacific/Easter,"<-06>6<-
05>,M9.1.6/22,M4.1.6/22"
Pacific/Efate,"<+11>-11"
Pacific/Enderbury,"<+13>-13"
Pacific/Fakaofo,"<+13>-13"
Pacific/Fiji,"<+12>-
12<+13>,M11.2.0,M1.2.3/99"
Pacific/Funafuti,"<+12>-12"
Pacific/Galapagos,"<-06>6"
Pacific/Gambier,"<-09>9"
Pacific/Guadalcanal,"<+11>-11"
Pacific/Guam,"ChST-10"
Pacific/Honolulu, "HST10"
Pacific/Kiritimati,"<+14>-14"
Pacific/Kosrae,"<+11>-11"
Pacific/Kwajalein,"<+12>-12"
Pacific/Majuro,"<+12>-12"
Pacific/Marquesas,"<-0930>9:30"
Pacific/Midway, "SST11"
Pacific/Nauru,"<+12>-12"
Pacific/Niue,"<-11>11"
Pacific/Norfolk,"<+11>-
11<+12>,M10.1.0,M4.1.0/3"
Pacific/Noumea,"<+11>-11"
Pacific/Pago_Pago,"SST11"
Pacific/Palau,"<+09>-9"
Pacific/Pitcairn,"<-08>8"
Pacific/Pohnpei,"<+11>-11"
Pacific/Port_Moresby,"<+10>-10"
Pacific/Rarotonga,"<-10>10"
Pacific/Saipan,"ChST-10"
Pacific/Tahiti,"<-10>10"
Pacific/Tarawa,"<+12>-12"
Pacific/Tongatapu,"<+13>-13"
Pacific/Wake,"<+12>-12"
Pacific/Wallis,"<+12>-12"
Etc/GMT,"GMT0"
Etc/GMT-0,"GMT0"
Etc/GMT-1,"<+01>-1"
Etc/GMT-2,"<+02>-2"
Etc/GMT-3,"<+03>-3"
Etc/GMT-4,"<+04>-4"
Etc/GMT-5,"<+05>-5"
Etc/GMT-6,"<+06>-6"
Etc/GMT-7,"<+07>-7"
Etc/GMT-8,"<+08>-8"
Etc/GMT-9,"<+09>-9"
Etc/GMT-10,"<+10>-10"
Etc/GMT-11,"<+11>-11"
Etc/GMT-12,"<+12>-12"
Etc/GMT-13,"<+13>-13"
Etc/GMT-14,"<+14>-14"
Etc/GMT0,"GMT0"
Etc/GMT+0,"GMT0"
Etc/GMT+1,"<-01>1"
Etc/GMT+2,"<-02>2"
Etc/GMT+3,"<-03>3"
Etc/GMT+4,"<-04>4"
Etc/GMT+5,"<-05>5"
Etc/GMT+6,"<-06>6"
Etc/GMT+7,"<-07>7"
Etc/GMT+8,"<-08>8"
Etc/GMT+9,"<-09>9"
Etc/GMT+10,"<-10>10"
Etc/GMT+11,"<-11>11"
Etc/GMT+12,"<-12>12"
Etc/UCT,"UTC0"
Etc/UTC,"UTC0"
Etc/Greenwich,'GMT0'
Etc/Universal,"UTC0"
Etc/Zulu,"UTC0"
```

## Program explanation

The program uses the following standard libraries.

```
//-----------------------------------------
// ESP32 Includes defines and initialisations
//-----------------------------------------

#include <Preferences.h>
                      #if ESP_ARDUINO_VERSION >= ESP_ARDUINO_VERSION_VAL(3, 0, 0)
#include "EdSoftLED.h"         // for LED strip WS2812 or SK6812
                      #else
#include <Adafruit_NeoPixel.h> // for LED strip WS2812 or SK6812
                      #endif
#include <NimBLEDevice.h>      // For BLE communication
#include <ESPNtpClient.h>      // https://github.com/gmag11/ESPNtpClient
#include <WiFi.h>              // Used for web page
#include <AsyncTCP.h>          // Used for webpage
#include <ESPAsyncWebServer.h> // Used for webpage  ESPAsyncWebServer
#include <ElegantOTA.h>
#include <ESPmDNS.h>
#include <Wire.h>
#include <RTClib.h>            // Used for connected DS3231
#include <Encoder.h>
#include <Keypad.h>            // For 3x1 membrane keypad instead of rotary encoder
```

The TAB in the IDE is the web page to display in the browser.
The #include "Webpage.h" a few lines further in the code loads the webpage.
I made the web page in the free 'Microsoft Expression Web 4'. It is not maintained anymore but has more than enough functionalities for our purposes.

To copy the code from the MS-Expression web HTML editor:
In the bottom line of the window of MS-Expression click 'Split'.
In the upper half the raw HTML Code is displayed and in the bottom half the Design window.
Copy in the Code window all the HTML code
Go to the Arduino IDE and paste it in the webpage.h TAB between:
R"rawliteral(  ... and ...  )rawliteral";

Or copy the code from the webpage.h into MSexpression Code area and redesign the page.

ESP32C3ClockAndMoreV.html* espinputformV52.html*

<body> <table.auto-style11> <tr> <td>

```
<!DOCTYPE HTML><html><head>
  <title>ESP32-C3 Word clock</title>
  <meta name="viewport" content="width=device-width, initial-scale=1
  <style type="text/css">
  .auto-style1 {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    background-color: #FFFFFF;
  }
  .auto-style2 {
      text-align: center;
  }
  .auto-style3 {
      font-family: Verdana, Geneva, Tahoma, sans-serif;
      background-color: #FFFFFF;
      color: #FF0000;
  }
```

td

**ESP32-C3 Word Clock**

Enter time as: hhmmss (132145)

**A** SSID **B** Password **C** BLE beacon

**D** Date (D15012021) **T** Time (T132145)

**E** Set Timezone E<-02>2 or E<+01>-1

Own colour: (FRRGGBB Hex:0-F)

**F** Font **G** Dimmed **H** Bckgnd

A long list if definitions and initialisations follows.
I am not a fan of passing all the variables to and from functions and like to keep them global in one program list.
If you write a program with other people it is good practice not to use too many global variables but this program is in one large listing, for the same reason to keep it simple.
I grouped all the variables per application to keep track where they are used.
With a simple find it is easy in this one great listing to find the back.

To print the time as text and colour with the proper LEDs or characters, the words and its position in a string of LEDs or text are defined.
The defines executes the function ColorLeds with its proper parameters.
Further in the program in the function void Dutch(void), void English(void) et cetera it becomes clear why these defines are so useful and handy.

```
 ...
#define PRECIES ColorLeds("precies", 16, 22, LetterColor);
#define MTIEN   ColorLeds("tien", 25, 28, LetterColor);
#define KWART   ColorLeds("kwart", 32, 36, LetterColor);
#define VOOR    ColorLeds("voor", 38, 41, LetterColor);
...
```

This is the initialisation of the storage area to store the struct EEPROMstorage.
The Struct with all its settings is saved in one command to permanent memory or SD Preferences FLASHSTOR;

struct EEPROMstorage { // Data storage to maintain them after power loss
byte DisplayChoice = 0;
...
char BLEbroadcastName[30]; // Name of the BLE beacon
char Timezone[50];
int Checksum = 0;
} Mem;

The variables are addressed with a short name Mem.
For example Mem.DisplayChoice = 3;

The Struct EEPROMstorage
is stored in the function StoreStructInFlashMemory
and retrieved in the function GetStructFromFlashMemory

The menu displayed in the serial monitor and BLE app is defined here.
String may not be longer than 40 characters what can be checked with the 'ruler' string.

```
// Menu
//0         1         2         3         4
//12345678901234567890123456789012345678901234567890
char menu[][40] = {
"A SSID B Password C BLE beacon name",
"D Date (D15012021) T Time (T132145)",
...
"W=WIFI, X=NTP, Y=BLE, Z=Fast BLE",
"Jun 2024" };
```

The Setup happens here:

```
//--------------------------------------------
// ARDUINO Setup
//--------------------------------------------
void setup()
{
 Serial.begin(115200);
// Setup the serial port to 115200 baud //
 pinMode(secondsPin, OUTPUT);
// turn On seconds LED-pin
...
...
msTick = millis();
}
```

The loop is kept almost empty and the program starts in EverySeccondCheck.
Only subroutine in CheckDevices(); must be executed often.

```
//--------------------------------------------
// ESP32 Loop
//--------------------------------------------
void loop()

{
Loopcounter++;                  // How often do the MCU loops/sec?
EverySecondCheck();
CheckDevices();
}
```

The following routines check if something must happen every second, minute, hour and day.
This flow handling of the program keeps the processor for 99% free for other uses.
In this program that is almost nothing but for other purposes this can be needed.
In the Bluetooth and Serial communication functions some short delays are used that are essential here but the program only runs here when there is an actual communication.
(An alternative method could have been the use of an interrupt every second and an empty loop).

```
//-------------------------------------------
// CLOCK Update routine done every second
//-------------------------------------------
void EverySecondCheck(void)
{
static int Toggle = 0;
 uint32_t msLeap = millis() - msTick;
//
 if (msLeap >999)           // Every second enter the loop
 {
  msTick = millis();
  GetTijd(false);           // Get the time for the seconds
  Toggle = 1-Toggle;        // Used to turn On or Off Leds
  UpdateStatusLEDs(Toggle);
  SetSecondColour();        // Set the colour per second of 'IS' and 'WAS'
  DimLeds(TestLDR);         // Every second an intensity check
  if (timeinfo.tm_min != lastminute) EveryMinuteUpdate();
  Loopcounter=0;
 }}
//-------------------------------------------
// CLOCK Update routine done every minute
//-------------------------------------------
void EveryMinuteUpdate(void)
{
...
if(timeinfo.tm_hour != lasthour) EveryHourUpdate();
}
//-------------------------------------------
// CLOCK Update routine done every hour
//-------------------------------------------
void EveryHourUpdate(void)
{
...
if (timeinfo.tm_mday != lastday) EveryDayUpdate();
}
// //
//-------------------------------------------------------------------------------
-----
// CLOCK Update routine done every day
//-------------------------------------------------------------------------------
-----
void EveryDayUpdate(void)
{
...
}

Check for serial input from the serial monitor and pass the command to
ReworkInputString()
//-------------------------------------------
// Common check for serial input
//-------------------------------------------
void SerialCheck(void)
{
```

```
...
ReworkInputString(SerialString+"\n"); // Rework ReworkInputString();
...
}
```

Restore all the default values.

```
//-------------------------------------------------------------------------
// Common Reset to default settings
//-------------------------------------------------------------------------
void Reset(void)
{
Mem.Checksum = 25065;
...
 }
```

Common print routines.
To keep all the print commands in one places it is easy to change these routines for other boards with a different 'slang'.

```
//-----------------------------------------
// Common common print routines
//-----------------------------------------
void Tekstprint(char const tekst[])
     { if(Serial) Serial.print(tekst); SendMessageBLE(tekst);sptext[0]=0; }
void Tekstprintln(char const tekst[])
     { sprintf(sptext,"%s\n",tekst); Tekstprint(sptext); }
void TekstSprint(char const tekst[])
     { printf(tekst); sptext[0]=0;} // printing for Debugging purposes
void TekstSprintln(char const tekst[])
     { sprintf(sptext,"%s\n",tekst); TekstSprint(sptext); }

//-------------------------------------------------------------------------
// Common Constrain a string with integers
// The value between the first and last character in a string is returned
between the low and up boundaries
//-------------------------------------------------------------------------
int SConstrainInt(String s,byte first,byte last,int low,int up)
    {return constrain(s.substring(first, last).toInt(), low, up);}
int SConstrainInt(String s,byte first, int low,int up)
    {return constrain(s.substring(first).toInt(), low, up);}
```

The setup of storage space and control of the validity of the settings.
In the checksum is invalid a reset() will restore the default settings

```
//-----------------------------------------
// Common Init and check contents of EEPROM
//-----------------------------------------
void InitStorage(void)
```

Store and retrieve the settings from SPIFFS or SD or EEPROM
the several possibilities are store here. EEPROM becomes outdated but still works.

```
-----------------------------------------
// COMMON Store mem.struct in FlashStorage or SD
//-----------------------------------------
void StoreStructInFlashMemory(void)
{
}
//-----------------------------------------
// COMMON Get data from FlashStorage Preferences.h
//-----------------------------------------
void GetStructFromFlashMemory(void)
{
}
```

Get the commands from the strings entered in the serial monitor, Bluetooth or the webpage and perform the command in an action.
The menu letters are almost used but it possible to distinguish between lower and uppercase when more commands are needed.
(That is why there is no conversion to UpperCase or LowerCase).

```
//-------------------------------------------
// CLOCK Input from Bluetooth or Serial
//-------------------------------------------
void ReworkInputString(String InputString)
{
....
switch (InputString[0])
{
case 'A':
case 'a':
if (InputString.length() >5 )
...
```

Read the LDR and divide it with 16 to get the values from 0 - 4096 between 0 and 255.
Not all boards has a 12 bit AD converter like.

```
//-------------------------------------------
// LDR reading are between 0 and 255.
// ESP32 analogue read is between 0 - 4096 --   is: 4096 / 8
//-------------------------------------------
int ReadLDR(void) {  return analogRead(PhotoCellPin)/16;}
```

Control the colour and intensity of the LED on the boards in one command

```
//-------------------------------------------
// CLOCK Control the LEDs on the ESP32
// 0 Low is LED off. Therefore the value is inversed with the ! Not
//-------------------------------------------
void SetStatusLED(bool Red, bool Green, bool Blue)
{
```

This function reads the analog port and calculates an output intensity to a display or LED-strip. The readings are squared to get a hyperbolic curve that resembles you eye correction for dark and light better than a linear range. It works wonderfully well.

```
//-------------------------------------------
// LED Dim the leds measured by the LDR and print values
// LDR reading are between 0 and 255. The Brightness send to the LEDs is
between 0 and 255
//-------------------------------------------
void DimLeds(bool print) { ... }
```

Here we print and colour the characters in the display or light up to proper LEDs in a String of RGB(W) LEDs.
The #define executes this functions with the proper parameters for every language and prints the texts in the serial connections.
#define QUARTER ColorLeds("quarter", 32, 38, LetterColor);

```
//-------------------------------------------    /
 / LED Set colour for LED.
```

```
//----------------------------------------
void ColorLeds(char const *Texkst, int FirstLed, int LastLed, uint32_t
RGBColor)
{  }
```

To convert all characters to uppercase in a character array.

```
//----------------------------------------
// COMMON String upper
//----------------------------------------
void to_upper(char* string)
```

Every display or strip uses other commands to regulate the brightness
Therefore for all LED/Display commands a function

```
//-------------------------------------------------------------------------
// LED Set brightness of backlight
//-------------------------------------------------------------------------
void SetBrightnessLeds(byte Bright)
{
 SetBackLight(Bright); // Set brightness of LEDs
}
```

A place to turn off all LEDs or clear the display

```
//----------------------------------------
// LED Clear the character string
//----------------------------------------
void LedsOff(void)
```

Here are all the colours are set for the characters are set.
The function has changed often and it's name describes it's original purpose
For backward compatibility it's name is unchanged.

```
//----------------------------------------
// LED Set second color
//----------------------------------------
void SetSecondColour(void)
{  switch (Mem.DisplayChoice)   {  case DEFAULTCOLOUR: LetterColor =
C_YELLOW;
...
```

SWversion() prints the menu and the settings of several preferences
The function has changed often and it's name describes it's original purpose
For backward compatibility it's name is unchanged.
PrintLine() prints the horizontal lines in the menu.

```
//----------------------------------------
// CLOCK Version and preferences info
//----------------------------------------
void SWversion(void)
{
#define FILENAAM (strrchr(__FILE__, '\\') ? strrchr(__FILE__, '\\') + 1 :
__FILE__)
PrintLine(35);
for (uint8_t i = 0; i < sizeof(menu) / sizeof(menu[0]);
Tekstprintln(menu[i++]));
...
PrintLine(35);
}
void PrintLine(byte Lengte)
{... }
```

Displaytime() prints the time to the serial monitor as text and control wiich language is printed.
It also sends the appropriate sequence of colour and intensities to a RGB(W) LED strip.

```
//-----------------------------------------
// CLOCK Say the time and load the LEDs
// with the proper colour and intensity
//-----------------------------------------
void Displaytime(void)
{
..
switch(Language) // Print all the character in the backgound color, a sort
of ClearScreen
{
case 0: strncpy(Template,"HETVISOWASOVIJFQPRECIESZSTIENKPFKWARTSVOORS
                         OVERAHALFSMIDDERTVIJFATWEESOEENOXVIERELFQTIENK
                         TWAALFBHDRIE CNEGENACHTFZESVZEVENOENVUUR",129);
        ColorLeds(Template,0,127, Mem.DimmedLetter);
        Dutch(); Print_tijd(); break;
case 1:
...
}
```

A series of functions to get and store time.
The NTP time server puts the retrieved time in the standard C time structures.

```
//--------------------------- Time functions --------------------------

void GetTijd(byte printit)
void Print_RTC_tijd(void)
void PrintNTP_tijd(void)
void PrintUTCtijd(void)
void Print_tijd(void)
void SetRTCTime(void)
```

Convert a HEX string to an unsigned 32-bits integer

```
//-----------------------------------------
// CLOCK Convert Hex to uint32
//-----------------------------------------
 uint32_t HexToDec(String hexString)
```

Functions to let the clocks speak the time in four languages
There is also a lot of slang in languages.
'Half nine' sometimes means 8:30 but can also be 9:30. (-:

```
/-----------------------------------------
// CLOCK Dutch clock display
//-----------------------------------------
void Dutch(void)
{
HET; // HET is always on
switch (timeinfo.tm_min)
{
case 0: IS; PRECIES; break;
case 1: IS; break;
case 2:
case 3: WAS; break;
case 4:
case 5:
...
```

```
void English(void)
void German(void)
void French(void)
```

The Bluetooth Low Energy Nordic nRF functions.
They are different from the Texas instrument CC2540/CC2541 that is used in other chipsets like the HM-10, HM16, JDY-08 et cetera.
More here on Instructables https://www.instructables.com/Communicate-Using-CC254x-or-NRF-BLE-With-Arduino-a/

```
/-----------------------------
// BLE SendMessage by BLE Slow in packets of 20 chars
//-----------------------------
void SendMessageBLE(std::string Message)
/-----------------------------
// BLE Start BLE Classes
//-----------------------------
class MyServerCallbacks: public BLEServerCallbacks

/-----------------------------
// BLE Start BLE Service
//-----------------------------
void StartBLEService(void)

/-----------------------------
// BLE CheckBLE
//-----------------------------
void CheckBLE(void)
```

Functions to start a WIFI connection and use the webpage

```
/-----------------------------------------
// WIFI WEBPAGE
//-----------------------------------------
void StartWIFI_NTP(void)

/-----------------------------------------
// WIFI WEBPAGE
//-----------------------------------------
void WebPage(void)

/-----------------------------------------
// WIFI WEBPAGE Not found message
//-----------------------------------------
void notFound(AsyncWebServerRequest *request)
```