

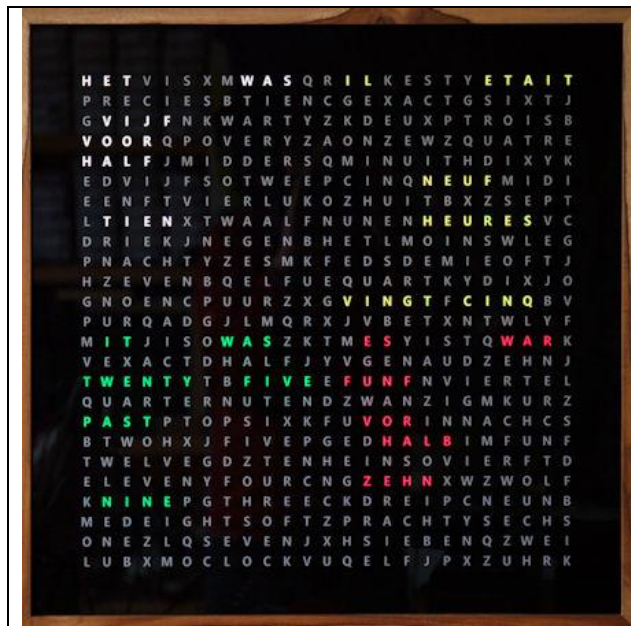
Arduino ESP32-Nano word clock

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

The Arduino ESP32 Nano is used to drive the clock.

Time is synchronized with the Network Time Protocol (NTP) from the internet.

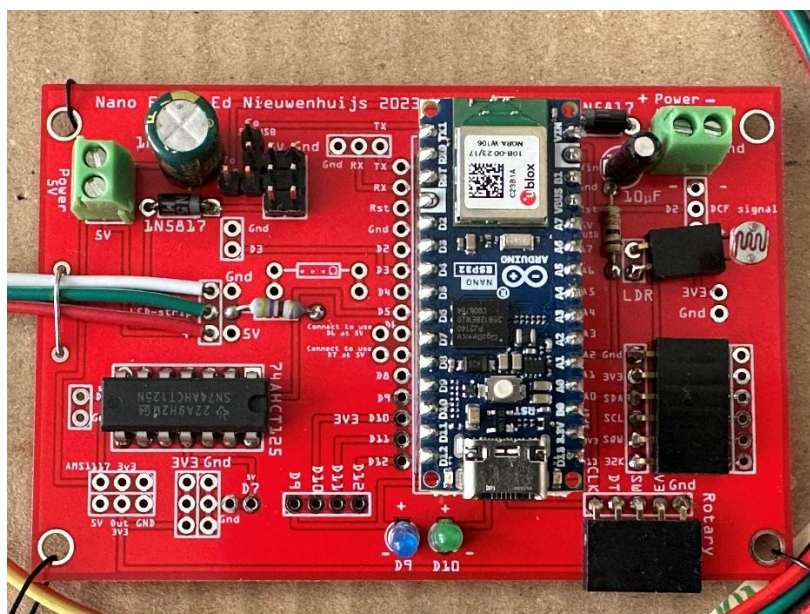
Settings can be controlled with a webpage, a PC or a Bluetooth Low Energy (BLE) serial terminal app installed on a phone, PC or tablet.



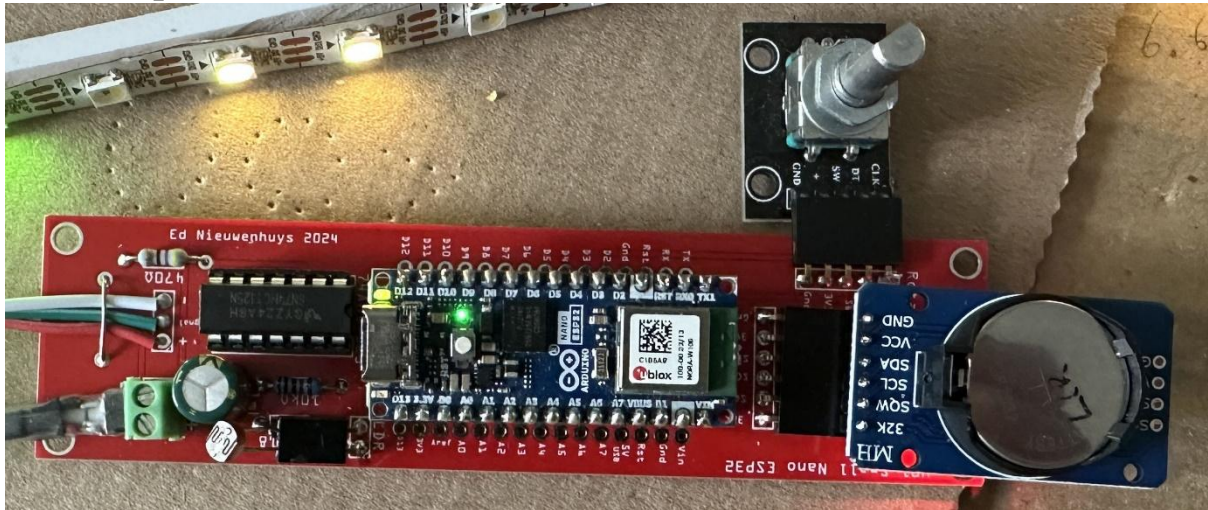
4-language clock



Dutch language clock



Arduino ESP32 Nano on the PCB inside the clock.



Small PCB design with Rotary and DS3231 RTC attached

Before starting

The clock receives time from the internet when there is a WIFI connection. When a DS3231 time module is attached to the circuit board an internet connection is not obliged. A rotary encoder or 3-button switch and the Bluetooth terminal app can be used to set time.

To connect to the internet the name of the WIFI station and its password must be entered in the clock software to be able to connect to a WIFI router. The name of the WIFI-station and password has to be entered once. These credentials will be stored in memory of the microprocessor. (See **Operation of the clock**)

To make life easy it is preferred to use a phone or tablet and a Bluetooth communication app to enter the WIFI credentials into the clock.

BLESerial nRF	BLE Serial Pro	Serial Bluetooth Terminal

- Download a Bluetooth UART serial terminal app on your phone, PC, or tablet.
 For IOS: BLE Serial Pro or BLESerial nRF.
 For Android: Serial Bluetooth Terminal.

First operation

- Plug in the power of the clock.

When there is no internet of Bluetooth connection the clock can be controlled with the rotary encoder or a three button switch.

- Press the rotary shaft or the middle button of the switch.
UUR will light up 3 times and the hours can be set by turning the shaft or pressing the up or down button.
- A second press on the shaft or middle button will blink HETISWAS and the minutes can be changed.
Seconds are set to 0. So to be exact turn the last minute exact at 0 seconds
- With a third press TWAALF will blink three times and the light intensity of the LEDs can be set
- With a fourth press the characters NTP and RTC will blink three times. One can switch from RTC to NTP time.
- Pressing nine times will Reset the clock to it factory settings and the clock will use the not very accurate clock that will drift seconds a day.

When the Bluetooth app is installed on your phone or tablet

- Start de Bluetooth terminal app on your phone or tablet
- Click on the connection named 'wordclock' in the serial terminal app.
- Find your WIFI router login credentials. Find the SSID of the router WIFI and its password. They are normally printed at the bottom of your WIFI router

Type in:

- aSSID and press Send.
SSID is the name of the SSID you can find for instance in your phone. It is the name WIFI is connected to.
- bPASSWORD and press send.
PASSWORD is the password you can find at the bottom of your router .
- cBLENAME
The BLENAME will be the name of your clock in the BLE serial terminal app and in the WIFI router connected.

Home Network Devices enabled: 16		
● FRITZ!WLAN Repeater 1750E	LAN	
● Ed-IPad-Pro	Wi-Fi - 5-GHz	
● FiboStick50cm	Wi-Fi - 2.4-GHz	
● FlipoKlok	Wi-Fi - 2.4-GHz	
● SteelClock-01	Wi-Fi - 2.4-GHz	

In the FRITZBox WIFI router
example above "SteelClock-01"

You can access the web page menu
of the clock by typing in a browser:
<http://SteelClock-01> or SteelClock-
01.local.

ESP32-Nano Word Clock

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

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

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

J Toggle use of DS3231 time module

U Demo mode (msec) (U200)

N Display Off between Nhhhh (N2208)

O Display toggle On/Off

P Status LED toggle On/Off

Display colour choice

Q0 Yellow **Q1** Hourly **Q2** White

Q3 All Own **Q4** Own **Q5** Wheel

F Own colour (Hex code FWWRRGGBB)

Light intensity setting (0-255)

S Slope **L** Min **M** Max

Communication

W WIFI **X** NTP **CCC** BLE

R Reset **@** Restart **+** Fast BLE

Self test **!** See RTC **&** Update RTC

[Manual](#) ednieuw.nl

Send

Web page in a browser

Building the clock

The building 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>

Build instruction of 4-language word clock with SK6812 LEDs in UK, NL, DE, FR with Nano. . <https://github.com/ednieuw/FourLanguageClock>

These build instruction are also used for clock with a Arduino Nano Every.

The rotary encoder and DS3231 RTC are optional and not necessary.
They are recommended if a clock cannot receive WIFI.

In the clock menu, the rotary encoder can be switched on or off.
Switch the option off if no rotary encoder is installed.
H Toggle use rotary encoder

With option J Toggle use DS3231 RTC module

The use of the attached DS3231 can be switched On and Off. When switched On the NTP updates will be turned Off

Turn NTP updates On again ,with option X NTP, when the DS3231 option is turned Off again otherwise the clock will drift minutes a day.

Compilation and uploading

For every board that can be programmed with the Arduino IDE a compiler is available. The compiler can be selected from the boards manager menu

For the Arduino Nano ESP32 two versions are available.

One from Arduino itself that uses core version 2. This board is safe to use.

Optional

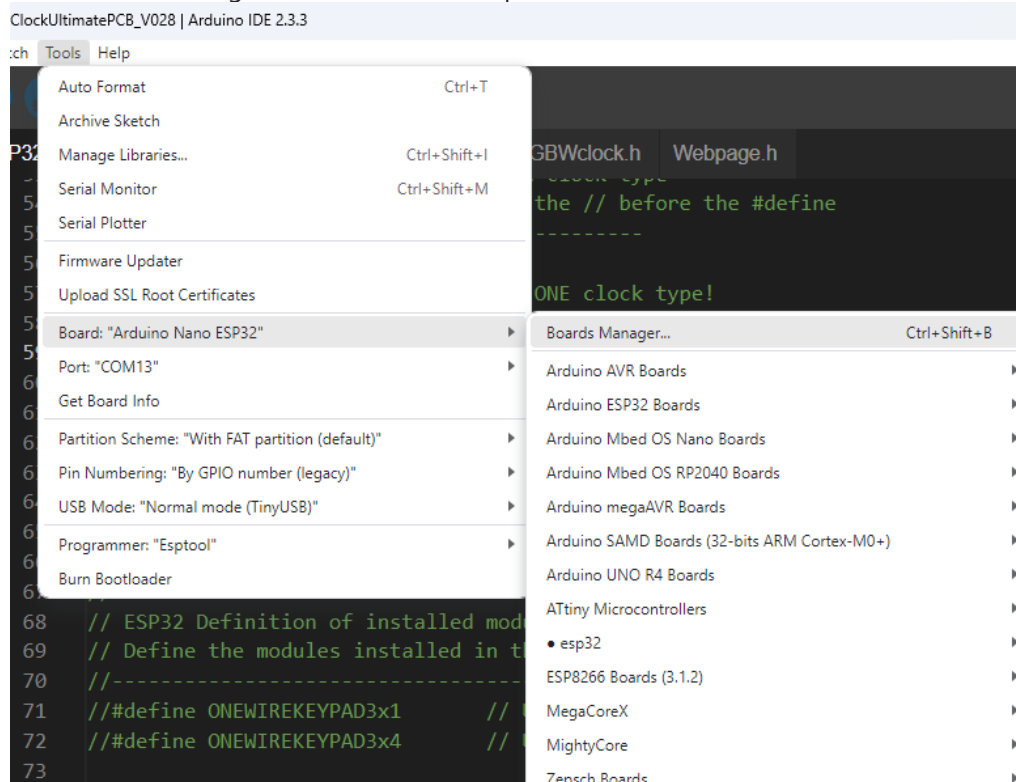
The manufacturer of the ESP32 microcontroller Espressif develops (in 2024) core version 3. Since version 3.0.5 this core compiles the word clock sketch V056.

But not with version 3.1.0

- Search for ESP32 in the board manager.

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

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



- Load the ESP32Arduino_WordClockV0xx.INO file in the IDE

Select one of the three word clocks

```
//#define FOURLANGUAGECLOCK
```

```
#define NL144CLOCK
```

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

- #define NL144CLOCK -> a 144 LED single language clock. Default language is Dutch. For French, German and English copy the coding from the four-language clock between the NL144CLOCK defines.

[Build instruction for the clock in Dutch and English](#) or [here in this repository](#)

ESP32Arduino_WordClockV061

- #define NLM1M2M3M4L114 -> a 114 LED single language clock with 4 extra LEDs for the minutes and a slightly different design
- #define FOURLANGUAGECLOCK -> a 4-language clock with 625 LEDs in a 25 x 25 grid.

[Build instruction of 4-language word clock with SK6812 LEDs in UK, NL, DE, FR with Nano.](#)

In the libraries.zip are the libraries to compile the software. Unpack them in your libraries folder.

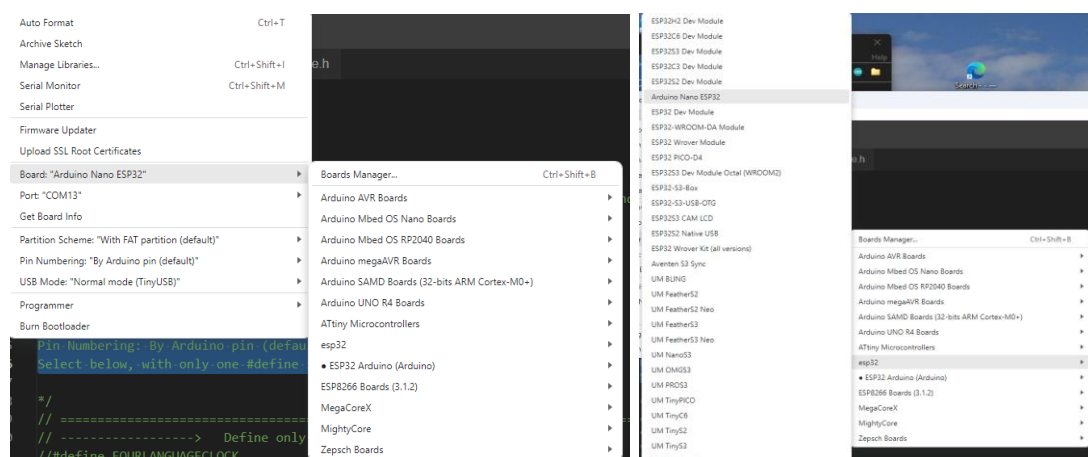
Board: Arduino Nano ESP32

Partition Scheme: With FAT

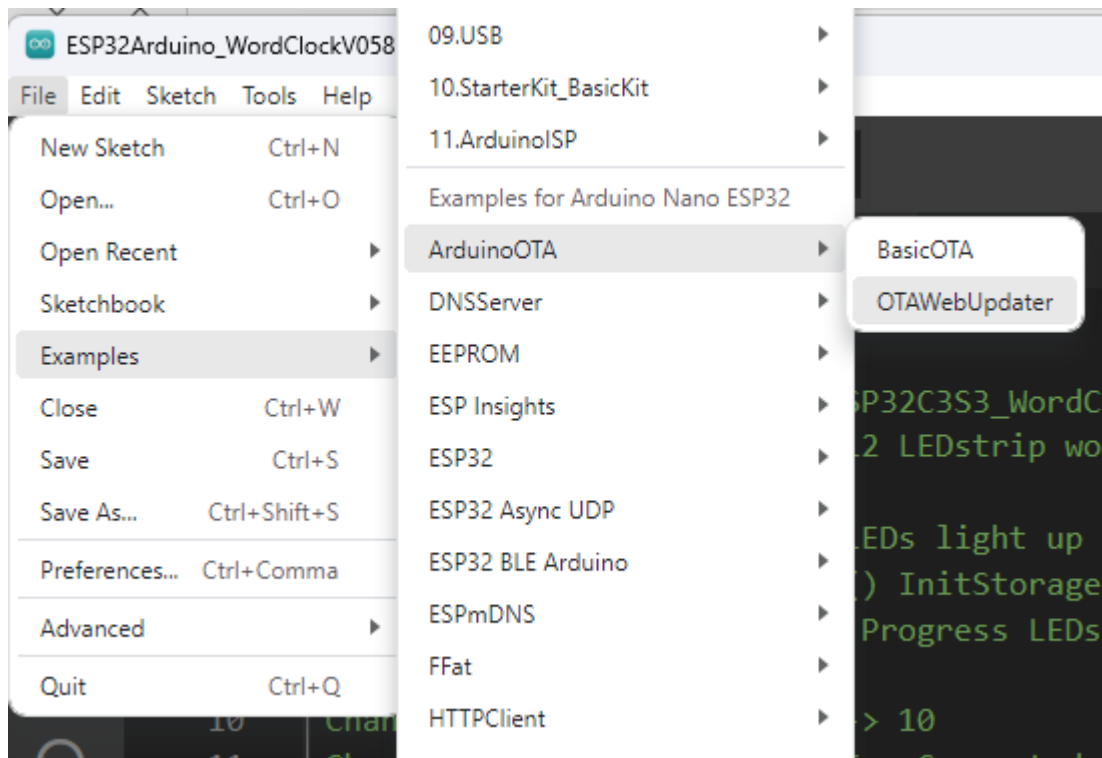
Pin Numbering: By GPIO number (legacy)

(if By Arduino pin (default) is chosen the LED-strip may not turn on and the RGB LED on the Nano ESP32 may not work)

If the LEDs do not turn on then it is most probably this pin settings that is wrong.



Alternative software installation



As described further on, software can also be placed on the Nano ESP32 'Over the Air'.

The advantage is that you do not 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 that is uploaded to the Nano ESP32 is saved as a .bin file.

Various .bin file versions of this software can be found on GitHub.

<https://github.com/ednieuw/Arduino-ESP32-Nano-Wordclock>

Short instructions

- Select the Nano ESP32 as board (Tools->Board-ArduinoESPboards->Arduino Nano ESP32)
 - Search in Examples for ArduinoOTA -> and open the program OTAWebUpdater
 - Choose Sketch->Upload or press the upload button at the top left
 - Open the serial monitor and see which IP-address is printed. (Press the white reset button on the Nano ESP32 if nothing is printed)
 - Type this IP-address into the URL of a browser (eg: 192.168. 0. 123)
 - Login with admin and admin
 - Find the .bin file and press update
- (For example 'ESP32Arduino_WordClockV057.inoNL114.bin' This is version V057 with settings for the NL114 clock).

Longer instructions

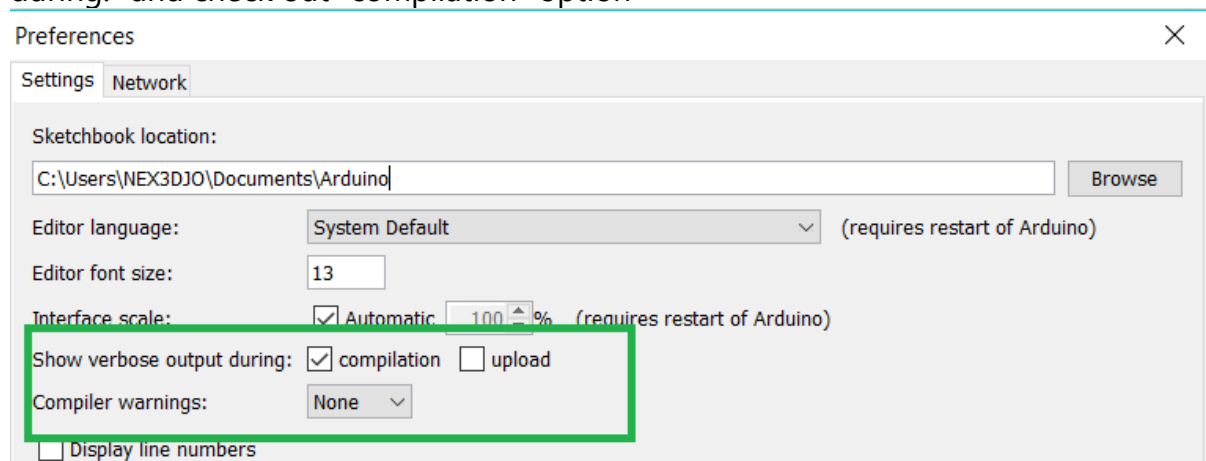
The sample implementation has been done using:

- Example sketch `OTAWebUpdater.ino`.
- ESP32 Board.

Before you begin, please make sure that you have the following software installed:

- Arduino IDE
- **Host software depending on O/S you use**
 - [Avahi](#) for Linux
 - [Bonjour](#) for Windows
 - Mac OSX and iOS - support is already built in / no any extra s/w is required

Prepare the sketch and configuration for initial upload with a serial port - Start Arduino IDE and load sketch OTAWebUpdater.ino available under File > Examples > OTAWebUpdater.ino - Update ssid and pass in the sketch so the module can join your Wi-Fi network - Open File > Preferences, look for "Show verbose output during:" and check out "compilation" option



- Upload sketch (Ctrl+U)
- Now open web browser and enter the url, i.e. <http://esp32.local>. Once entered, browser should display a form

ESP32 Login Page

 Username:
 Password:

- username = admin
- password = admin

Note

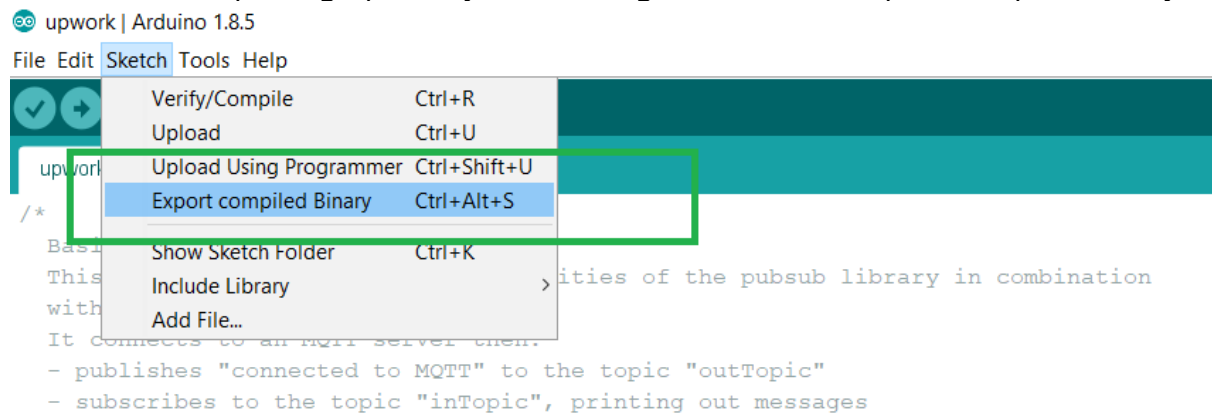
If entering "http://ESP32.local" does not work, try replacing "ESP32" with module's IP address. This workaround is useful in case the host software installed does not work.

Now click on the Login button and browser will display an upload form

Choose File No file chosen Update

progress: 0%

For Uploading the New Firmware, you need to provide the Binary File of your Code. Exporting Binary file of the Firmware (Code) - Open up the Arduino IDE - Open up the Code, for Exporting up Binary file - Now go to Sketch > export compiled Binary



- Binary file is exported to the same Directory where your code is present

Once you are comfortable with this procedure, go ahead and modify OTAWebUpdater.ino sketch to print some additional messages and compile it. Then, export the new binary file and upload it using web browser to see entered changes on a Serial Monitor.

Operation of the clock

To connect to a WIFI network a SSID (WIFI name) and password must be entered.

There are a few methods:

Connect the MCU in the clock with a micro USB serial cable to a PC and use a serial terminal.

Use a BLE serial terminal app on a phone or tablet for connection.

For a PC the app [Termite](#) is fine as serial terminal. But serial monitor in the Arduino IDE works often better.

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.

- Start the app and start a connection with the clock. Some apps automatically start with a connection window but for some a connection symbol must be pressed. There is most probably one station, the word clock, to select from.

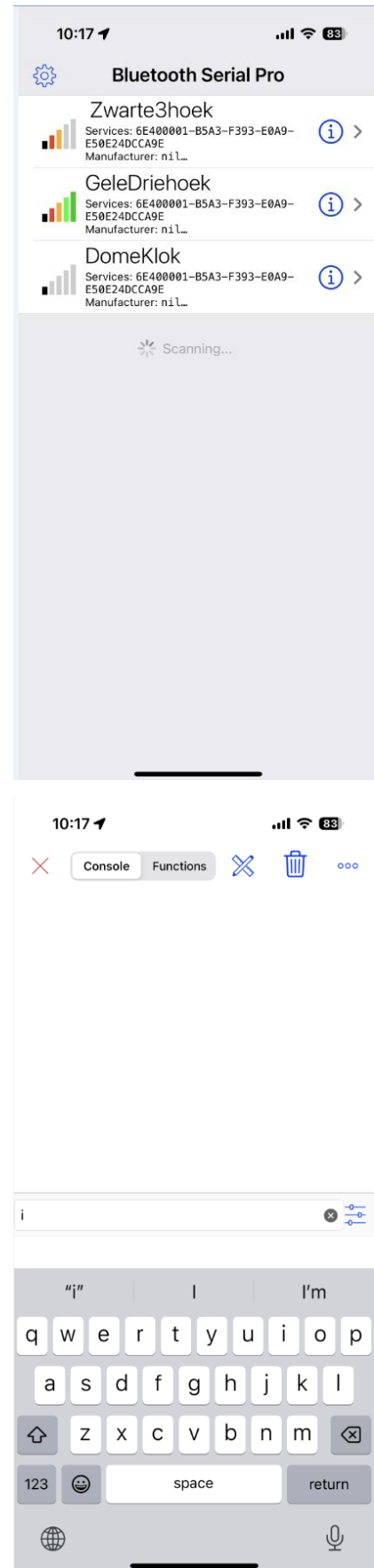
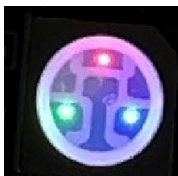
- Select the clock in the list.

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

- Sending the letter l or i for information will display the menu followed with the actual settings of several preferences.

In the clock there is a LED that will have a red dot lighted when the program is running.

A green dot will turn on when there is a WIFI connection. When there is a Bluetooth connection a blue dot in the LED will light.



In both cases **send the letter I of Information and the menu shows up.**

Enter the first letter of the setting to change followed with a code.

Some entries just toggle On and Off. Like the W to set WIFI Off or On.

To change the SSID and password:

Send the letter **A** or **a** followed with the WIFI station name.

Amy-ssid and send this command. Eg AFRITZ!Box01 or aFRITZ!Box01. Starting with an upper or lower case character is an identical instruction in the command string

Then the letter B followed with the password.

Bmypassword and send the password.

Cbroadcastname will change to name displayed in the Bluetooth connection list. Something like: cMyClock

If the length of the SSID and/or password is less than 5 characters the WIFI will be turned off automatically to avoid connection errors.

Use a length of minimal 8 characters for SSID and password.

Check in the menu (third row from the bottom) if WIFI and NTP are on.

Enter @ to restart the MCU. It will restart and connections will be made.

Sometimes a second or third reset must be given to get the clock connected to WIFI. If connection still fails check the SSID name and the entered password.)

If WIFI is connected the LED on the MCU will blink with a green dot.

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.

Default the clocktime zone is set to Amsterdam time.




A reset with option R in the menu will restore this time zone to Amsterdam again.

To set a different time zone send the time zone string between the quotes prefixed with the character E or e.

At the bottom of this manual many time zones are printed.

For example; if you live in Australia/Sydney send the string, eAEST-10AEDT,M10.1.0,M4.1.0/3.

The clock will use the Daylight saving time (DST) when connected to an NTP server but not when using the DS3231 time module

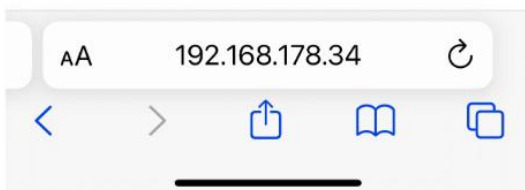
15:58   

ESP32-Nano 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
L Language **L0** NL **L1** UK
L2 DE **L3** FR **L4** Wheel
N Display off between Nhhhh (N2208)
O Display On/Off
P Own colour design (0-F) (P00FF00)
Q Display colour choice (Q0-Q6)
Q0 Yellow **Q1** Hourly **Q2** White
Q3 All Own **Q4** Own **Q5** Wheel
Q6 Digital **Q7** Analog -
R Reset settings @ Restart MCU
W WIFI On/Off **X** NTP &Requary
Y BLE On/Off **Z** Fast BLE

Send



HTML page on iPhone

```
Termite 3.4 (by CompuPhase)

A SSID B Password C BLE beacon name
D Date (D15012021) T Time (T132145)
E Timezone (E<-02>2 or E<+01>-1)
  Make own colour of: (Hex RRGGBB)
F Font  G Dimmed font H Bkgnd
I To print this Info menu
K LDR reads/sec toggle On/Off
L L0 = NL, L1 = UK, L2 = DE
  L3 = FR, L4 = Wheel
N Display off between Nhhhh (N2208)
O Display toggle On/Off
P Status LED toggle On/Off
Q Display colour choice (Q0-7)
  Q0 Yellow  Q1 hourly
  Q2 White   Q3 All Own
  Q4 Own     Q5 Wheel
  Q6 Digital Q7 Analog display
R Reset settings @ = Reset MCU
--Light intensity settings (1-250)--
S=Slope V=Min U=Max (S80 V5 U200)
W=WIFI X=NTP& Y=BLE Z=Fast BLE
Ed Nieuwenhuys Aug 2023

Display off between: 00h - 00h
Display choice: Yellow
Slope: 80      Min: 5      Max: 255
SSID: FRITZ!BoxEd
BLE name: PaarseS3N8R2
IP-address: 192.168.178.106 (/update)
Timezone:CET-1CEST,M3.5.0,M10.5.0/3
WIFI=On NTP=On BLE=On FastBLE=Off
Language choice: Rotate language
Software: ESP32C3S3_WordClockV024.ino
00/01/1900 00:00:00
```

Termite Terminal from a PC

Control and settings of the clock

The clock can be controlled 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 numbers and dots (for example: 192.168.178.77) must be entered in the web browser of your mobile or PC or instead use the BLE broadcast name followed with .local as URL in the browser. (Default after a reset: wordclock.local)

Or use a Bluetooth connection:

- Open the BLE terminal app.
- Look in the app for the clock to connect to and connect.

Every app has its own way of showing the Bluetooth device to connect to.

The iPhone/iPad/iMac BLE serial apps are made by me and work with their default settings.

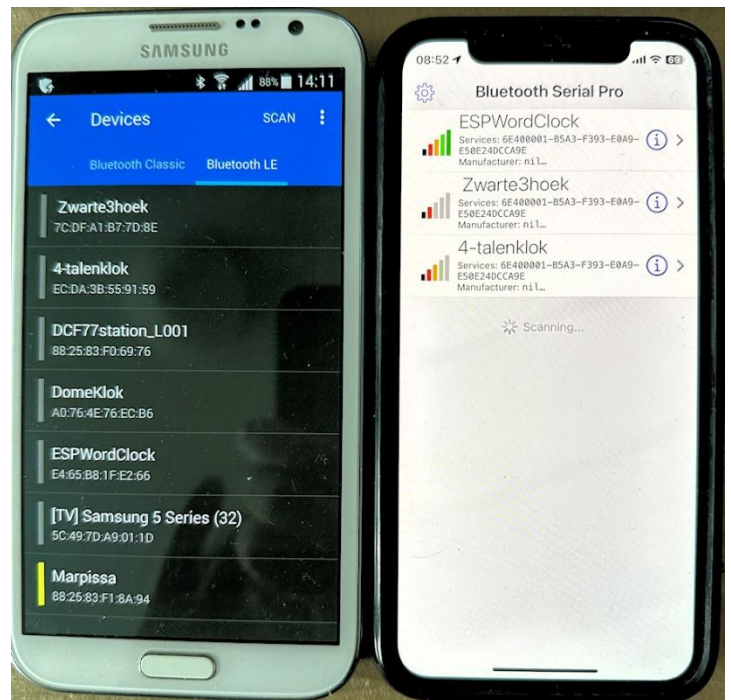
With other apps settings for sending and receiving data may have to be changed. Play with the font size and the CR and LF setting until you get On a iPhone, iPad or iMac with the BLE serial app it is possible to speed up the transmission speed by selecting option '+ Fast BLE' in the menu

Unfortunately some apps can not read strings longer than 20 characters and you will see the strings truncated or garbled.

If you see a garbled menu enter and send the character '+' to select the slower transmission mode.

If transmission is too garbled and it is impossible to send the character Z try the web page of the clock and send the character +.

If all fails you have to connect the MCU inside the clock with a USB-C cable to a PC and use a serial terminal app to send a + or as a last resort R to reset to factory settings.



Adjusting the light intensity of the display

In the menu 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 shown between the ().

S How fast the brightness reaches maximum brightness.

L How bright the display is in complete darkness.

M the maximum brightness of the display.

In the bottom half of the menu the values are displayed

Slope: 80 Min: 5 Max: 255

The clock reacts on light with its LDR (light dependent resistor).



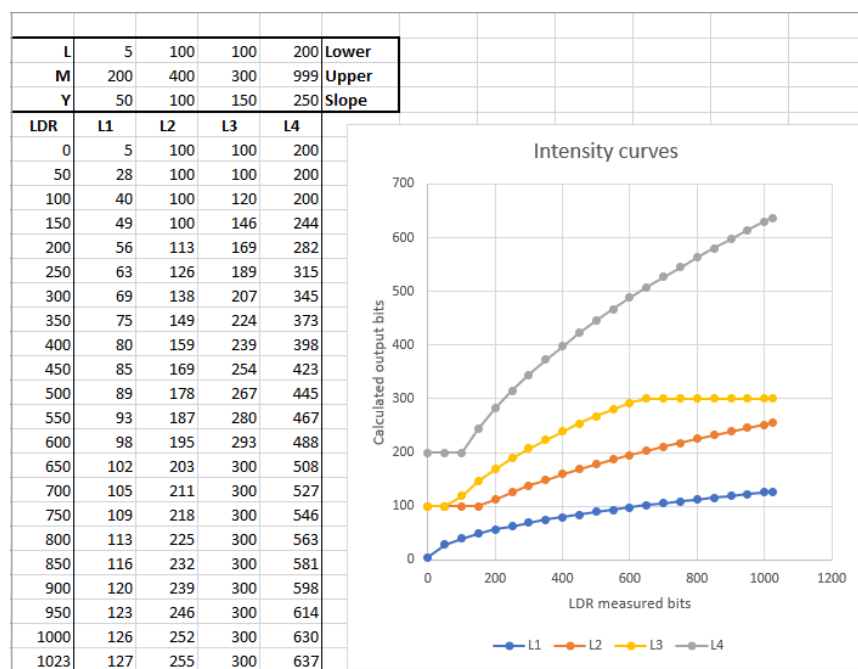
When it gets dark the display does not turn off completely but will stay dimmed at a minimum value.

With the parameter V the lowest brightness can be controlled. With a value between 0 and 255 this brightness can be set.

V5 is the default value.

The maximum brightness of the display is controlled with the parameter U. Also a value between 0 and 255.

With parameter S the slope can be controlled how fast maximum brightness is achieved.



Settings are set by entering the first character of a command following by parameters if necessary.

For example:

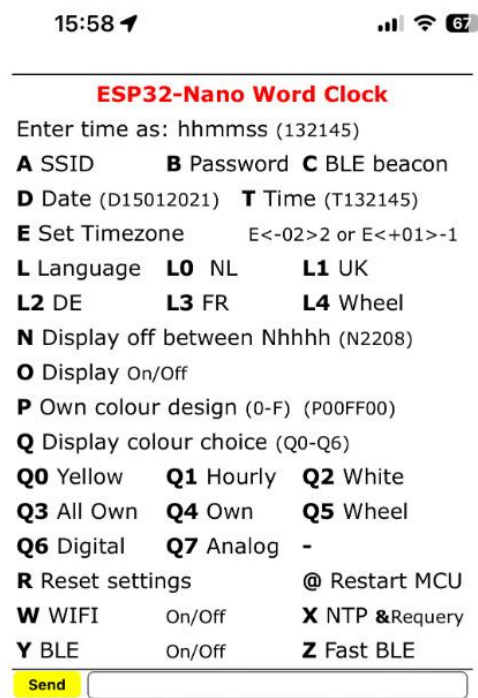
To set the colours of the words in the clock to white enter: Q2

To shown random all four languages every minute send L4. (This will not function in a single language clock)

Turn off WIFI by sending a W.

Restart the clock with the letter @.

Reset to default setting by send R.



15:58

ESP32-Nano 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

L Language **L0** NL **L1** UK

L2 DE **L3** FR **L4** Wheel

N Display off between Nhhhh (N2208)

O Display On/Off

P Own colour design (0-F) (P00FF00)

Q Display colour choice (Q0-Q6)

Q0 Yellow **Q1** Hourly **Q2** White

Q3 All Own **Q4** Own **Q5** Wheel


Q6 Digital **Q7** Analog -

R Reset settings **@** Restart MCU

W WIFI On/Off **X** NTP &Requery

Y BLE On/Off **Z** Fast BLE

Send



15:59

ESP32-Nano Word Clock

Enter time as: hhmmss (132145)

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

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

E Timezone (E<-02>2 or E<+01>-1)

Make own colour of: (Hex RRGGBB)

I To print this Info menu

K LDR reads/sec toggle On/Off

L L0 = NL, L1 = UK, L2 = DE

L3 = FR, **L4** = Wheel

N Display off between Nhhhh (N2208)

O Display toggle On/Off

P Status LED toggle On/Off

Q Display colour choice (Q0-Q7)

Q0 Yellow **Q1** hourly

Q2 White **Q3** All Own

Q4 Own **Q5** Wheel

Q6 Digital **Q7** Analog display

R Reset settings @ = Reset MCU

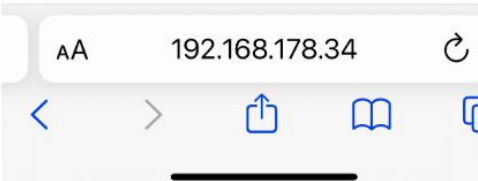
---Light intensity settings (1-250)---

S=Slope **V**=Min **U**=Max (S80 V5 U200)


W=WIFI **X**=NTP& **Y**=BLE **Z**=Fast BLE

Ed Nieuwenhuys April 2024

Send



AA 192.168.178.34

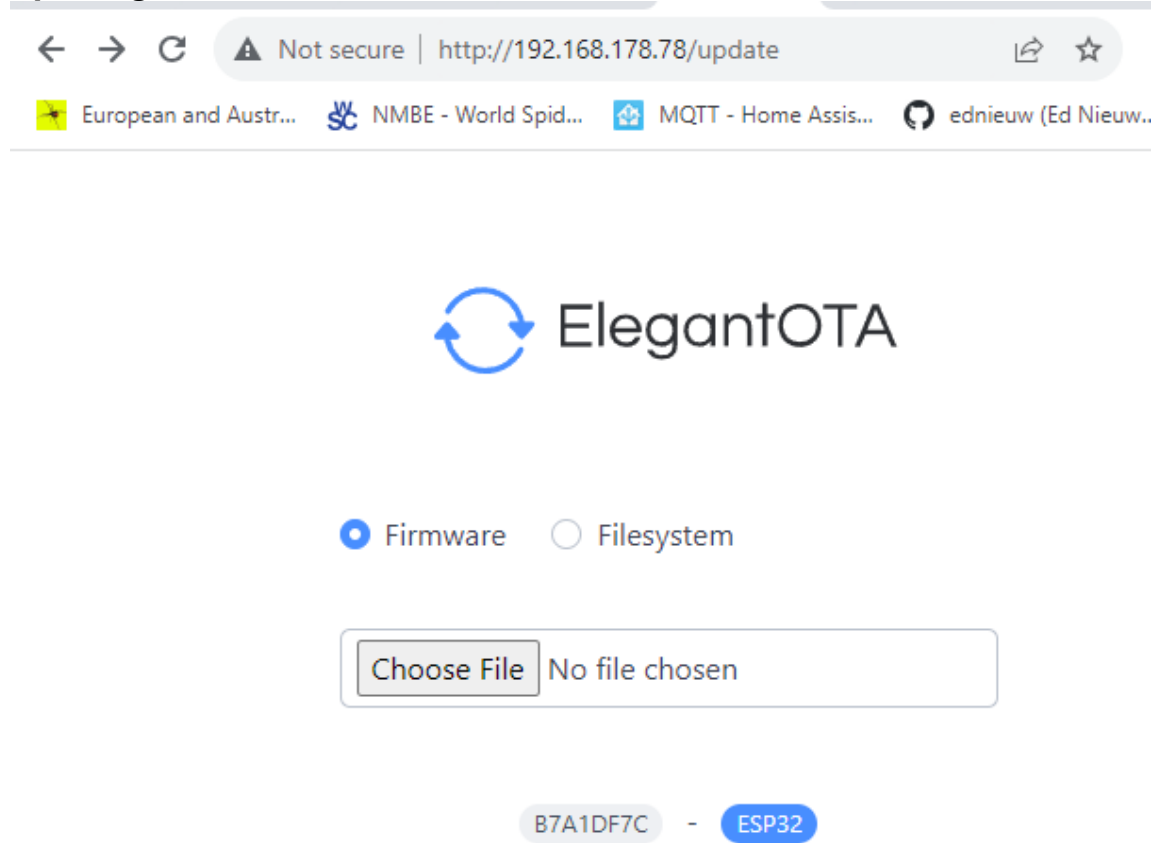


Enter Message

HTML page

BLE menu

Updating the software



The screenshot shows a web browser window with the address bar displaying 'http://192.168.178.78/update'. The page title is 'Updating the software'. The main content area features the 'ElegantOTA' logo, which consists of a blue circular arrow icon and the text 'ElegantOTA'. Below the logo, there are two radio buttons: 'Firmware' (selected) and 'Filesystem'. A file selection box is present with a 'Choose File' button and the text 'No file chosen'. At the bottom, there is a label 'B7A1DF7C' followed by a minus sign and a blue button labeled 'ESP32'.

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

Enter the IP-address of the clock followed with /update.

Something like this : 192.168.178.141/update.

Or use the BLEbroadcastname.local/update instead of the IP-address.

In this case: redpcbv01.local/update.

```
# Self test, ! See RTC, & Update RTC
Ed Nieuwenhuys November 2024

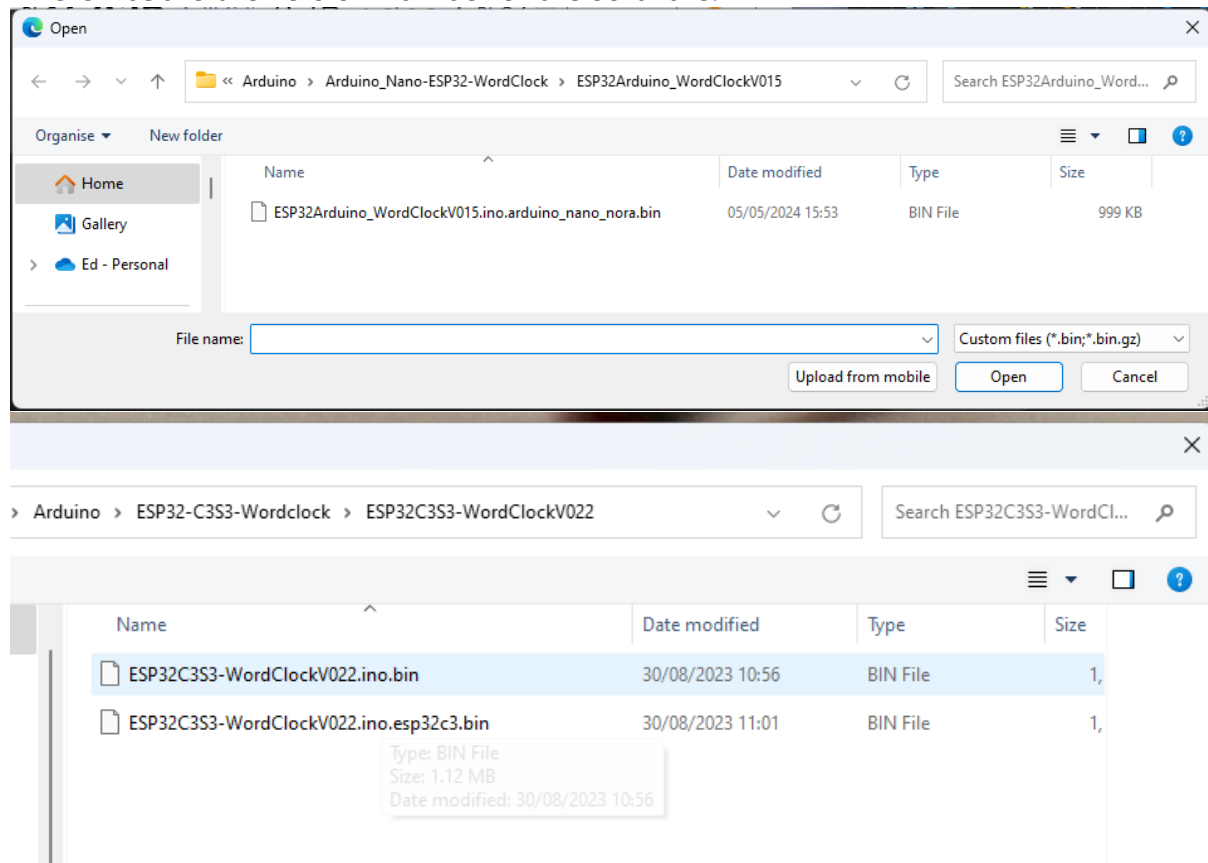
Display off between: 23h - 08h
Display choice: Yellow
Slope: 10      Min: 5      Max: 255
SSID: FRITZ!BoxEd
BLE name: RedPCBV01
IP-address: 192.168.178.141 (/update)
Timezone: CET-1CEST,M3.5.0,M10.5.0/3
```

'Choose File' in the menu and select the bin file to update.

Something like: ESP32Arduino_WordClockV056.ino.bin

ESP32Arduino_WordClockV061

where V056 is the version number of the software.



Detailed description

With the menu many preferences can be set. These preferences are stored in the ESP32 storage space.

Enter the first character in the menu of the item to be changed followed with the parameter.

There is no difference between upper or lower case. Both are OK.

```
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
```

ESP32Arduino_WordClockV061

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 Toggle use rotary encoder
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 Digital
R Reset settings @ = Reset MCU
U Demo mode (msec) (M200)
--Light intensity settings (1-250)--
S Slope, L Min, M Max (S80 L5 M200)
W WIFI, X NTP, CCC BLE, + Fast BLE
Self test, ! See RTC, & Update RTC
Ed Nieuwenhuys November 2024

Display off between: 23h - 08h
Display choice: Yellow
Slope: 5 Min: 5 Max: 100
SSID: FRITZ!BoxEd
BLE name: RedSmall101
IP-address: 192.168.178.199 (/update)
Timezone: CET-1CEST, M3.5.0, M10.5.0/3
WIFI=On NTP=Off BLE=On FastBLE=On
LED strip: SK6812 (Send % to switch)
Software: ESP32Arduino_WordClockV049.ino
ESP32 Arduino core version: 2.0.17
01/01/2000 00:06:42

Het is vijf over twaalf
Het is vijf over twaalf LDR:140= 3% 220692 1/s 20°C 00:06:43
Het was vijf over twaalf LDR:140= 3% 215629 1/s 20°C 00:07:00

A SSID B Password C BLE beacon name

Enter the name of the SSID and its password of the router the clock will to be connected to with option A and B.

Change with option C the Bluetooth broadcast name that will be displayed in the Bluetooth serial terminal app and in the WIFI router the clock is connected to.

For example: **a**FRITZ!Box or **A**FRITZ!Box.

Then enter the password. For example: **B**Secret_password.

Optionally enter **c**BluetoothName (default after a reset: wordclock)

Restart the clock by sending @.

CCC Entering CCC or ccc will toggle BLE on or off. Be careful 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 not connected to WIFI time and a RTC DS3231 is attached date must be set by hand.

For example enter: D06112022 to set the date to 6 November 2022.

Enter for example T132145 (or 132145 , or t132145) to set the time to 45 seconds and 21 minute past one o'clock.

Changing date and time only works when WIFI and NTP are off.

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

At the bottom of this page the time zones used in 2022 can be found.

It is a rather complicated string and it is therefore wise to copy it.

Let's pick one if you happen to 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 with starting with an 'E' or 'e' in front.

E<+00>0<+02>-2,M3.5.0/1,M10.5.0/3

F Own colour (Hex WWRRGGBB)

The colours of your choice of the words displayed in the clock 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 is 2 digits for Red followed with two for Green and ending with two digits for Blue.

To colour the characters intense Green enter F000FF00 prefixed with the letter F.

To colour the characters intense red enter FF0000 prefixed with the character F.

To set intense blue enter: F000FF or FFF.

To set the dimmed character to dark gray enter for example: F191919.

With SK6812 LEDs the extra white LED can be used besides the three RGB LEDs in the same housing.

For example: F8800FF00 is 50% white with 100% green.

G Scan for WIFI networks in the neighbourhood

Sending 'G' will print the available networks in the neighbour hood in the serial terminal. This can be easy to copy the sometimes difficult SSID into option A of the menu.

H H001 rotary, H002 membrane (H000)

Sending 'Hnnn' will select an optional rotary encoder or 3-button (membrane) switch attached to the PCB.

Turned on and not attaching a rotary encoder will probably result in unintentional input because the not attached pins will pick up signal randomly.

After a reset H000, nothing attached, will be selected

I To print this Info menu

Print the menu to Bluetooth and the serial monitor connected with an USB-cable.

J Toggle to use the DS3231 module time

Sending 'J' will toggle the use of an optional DS3231 time module ON and OFF.

If the clock does not has an internet connection time will probably drift undesirably quick. Installing a DS3231 time module will reduce the drift to a few seconds per year. Time can be entered with option T and D in the menu.

K Reads/sec toggle On/Off

Entering a K toggles printing of the LDR reading of the measured light intensity.

It also shows how many times the processor loops through the program and checks its tasks to run the clock.

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

N Display off between Nhhhh (N2208)

With N2208 the display will be turned off between 22:00 and 8:00.

O Display toggle On/Off

O Toggles the display off and on.

P Status LEDs toggle On/Off

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

Q Display colour choice (Q0-6)

Q0 Yellow Q1 hourly Q2 White Q3 All Own Q4 Own Q5 Wheel Q6 Digital display

Q0 will show the time with yellow words

and will change HET)or IT) from green to red via yellow in an hour.

and will change IS or WAS from green to red via yellow in an minute.

Q1 will show every hour another colour.

Q2 shows all the texts white.

Q3 and Q4 uses your own defined colours.

Q5 will follow rainbow colours every minute.

Q6 is the digital display

R Reset settings

R will set all preferences to default settings.

It also clears the SSID and password.

Perform a reset when the sketch is uploaded for the first time in the Arduino Nano ESP32.

S=Slope L=Min M=Max (S50 L5 M200)

S How fast 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 250'

U Demo mode (msec) (U200)

Enter U followed with the duration of a second in milli seconds.

M200 (200 milli second) will speed up the clock 5 times.

Sending an U will turn off the demo mode.

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

Toggle WIFI, NTP on and off.

Enter the character will toggle it on or off.

At the bottom of the menu the stated is printed.

```
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
0 1 2 3 4 5 6 7 8 9 A B C D E F
```

Sending a & will start a query from the time server.

+ Fast BLE

The BLE UART protocol sends default packets 20 bytes long. Between every packet there is a delay of 50 msec.

The IOS BLEserial app, and maybe others too, is able to receive packets of 80 bytes or more before characters are missed. This makes the menu printed faster.

Option + toggles between 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 and being used to show a realistic time.

Same as & option but this option will not update from the internet NTP server but only shows the time.

= Selftest

Sending a # will start the clock self-test. This is convenient to check if all the words in the clock are functioning.

% = Switch 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 a R in the menu will not change the LED strip selection.

@ = Reset MCU

@ will restart the MCU. This is handy when the SSID, et cetera are changed and the program must be restarted. Settings will not be deleted.

& = Get and stores NTP time in RTC and DS3231 time

& will get the NTP time immediately from the internet and stores it in the RTC clocks. This option is convenient to force the clock to get the proper NTP time.

In other cases the program will check the time running in the clock and on the NTP server so now and then and update the RTC clocks.

The DS3231 time module must be installed and being used to show a realistic time.

123456 Set time in RTC module

ESP32Arduino_WordClockV061

Enter the time as 152300 hhmmss. Same as T152300

Changing date and time only works when WIFI and NTP are off.

Program explanation

The program uses the following standard libraries.

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

#include <Arduino.h>
#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 <Wire.h>
#include <RTCLib.h> // Used for connected DS3231
#include <Encoder.h>
```

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'.

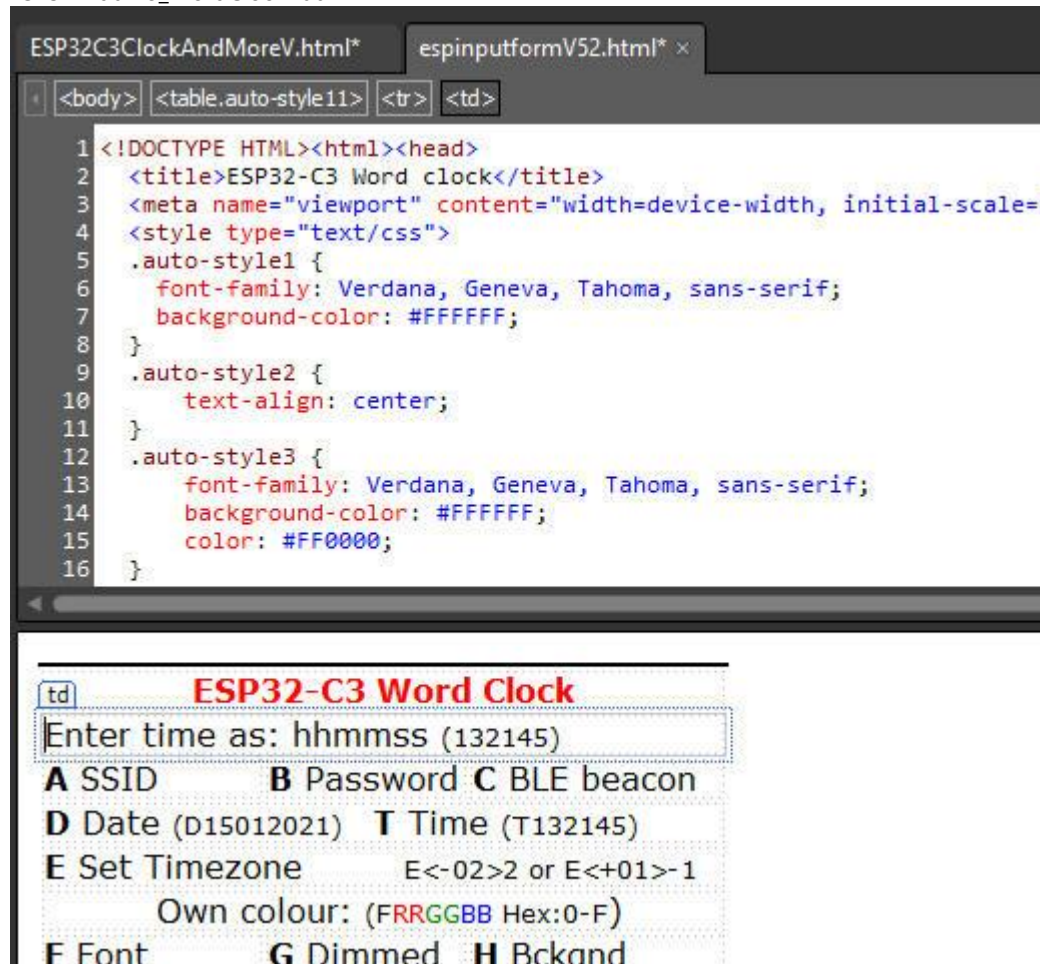
In the upper half the raw HTML Code is displayed and in the bottom half the Design window.

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.



A long list of 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;

ESP32Arduino_WordClockV061

```
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
//1234567890123456789012345678901234567890
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 EverySecondCheck.
Only subroutine in CheckDevices(); must be executed often.

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

ESP32Arduino_WordClockV061

```
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()

ESP32Arduino_WordClockV061

```
//-----  
// 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)  
{  
}
```

ESP32Arduino_WordClockV061

```
//-----  
// 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

ESP32Arduino_WordClockV061

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)
{
```

ESP32Arduino_WordClockV061

```
#define FILENAAM (strchr(__FILE__, '\\') ? strchr(__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 which 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 background color, a sort  
of ClearScreen  
{  
case 0: strncpy(Template,"HETVISOWASOVIJFQPRECIESZSTIENKPFKWARTSVOORS  
OVERAHALFSMIDDERTVIJFATWEESOENOXVIERELFQTIENK  
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 a 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
```



```

ESP32Arduino_WordClockV061
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/) <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)

```

Time zones

Copy the text **between the quotes** and paste them after the character E and send it to the clock

For example EGMT0.

```
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,"WAT-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,"WAT-1"
Africa/Lagos,"WAT-1"
Africa/Libreville,"WAT-1"
Africa/Lome,"GMT0"
Africa/Luanda,"WAT-1"
Africa/Lubumbashi,"CAT-2"
Africa/Lusaka,"CAT-2"
Africa/Malabo,"WAT-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,"HST10HDT,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"
```

ESP32Arduino_WordClockV061

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"

ESP32Arduino_WordClockV061

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"

ESP32Arduino_WordClockV061

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-10AEDT,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, "WIT-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"

ESP32Arduino_WordClockV061

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"

ESP32Arduino_WordClockV061

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/Bougainville, "<+11>-11"
Pacific/Chatham, "<+1245>-12:45<+1345>,M9.5.0/2:45,M4.1.0/3:45"

ESP32Arduino_WordClockV061

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"

ESP32Arduino_WordClockV061
Etc/Universal,"UTC0"
Etc/Zulu,"UTC0"

