

Bouwinstructies 12x12 LEDs RGB(W) woordklok

[Terug naar Start](#)

Deze woordklok gebruikt SK6812 RGBW om de tekst van de klok te in een kleur te verlichten.
De letters van de woordplaat vormen verlichte woorden die de tijd weergeven.

Deze woordklok met een ESP32 wordt beschreven [hier op Github](#).

RGBW LED-strips hebben naast de RoodGroenBlauw-LED een witte LED ingebouwd.

Hierdoor is het mogelijk om naast gekleurde letters ook helder witte woorden te zien.

De software voor deze klok heeft de optie om WS2812 RGB LEDs te gebruiken.

Met RGB LEDs heeft wit een lichte kleurzweem.

De LED's bevatten ieder een kleine processor waardoor de hele keten van LEDs met één draad is aan te sturen.

De software voor deze klok is ontworpen voor 12 x 12 LEDs in een kast van 25 of 50 cm.

LED-strips zijn te koop met 30 of 60 LEDs per meter zodat er precies één LED achter elke letter zit en de strip niet verknipt hoedt te worden.

Dat maakt de bouw eenvoudiger en biedt de mogelijkheid voor een digitale klok display.

De [witte LED-klok](#) is ook te overwegen.

De software source bevat code voor meerdere opties in uitvoering van de klok

Met #defines in de software worden de diverse opties aan of uit gezet.

De eerste drie modules zijn in ieder geval nodig om de klok te gebruiken

- RTC DS3231 ZS-042 clock module
- KY-040 Keyes Rotary Encoder
- LDR light sensor GL5528

Bijzonder aan te raden is

- Bluetooth om de klok met [iOS BLE Serial Pro](#) of Android app aan te sturen.

Extra code voor:

- DCF77 module DCF-2
- LCD
- WIFI on MKR 1010 NTP time
- HC12 transceiver
- 3x4 of 3x1 membraan toetsenbord

- [Vier talen](#); Nederlands, Duits Frans en Engels

- Een handleiding

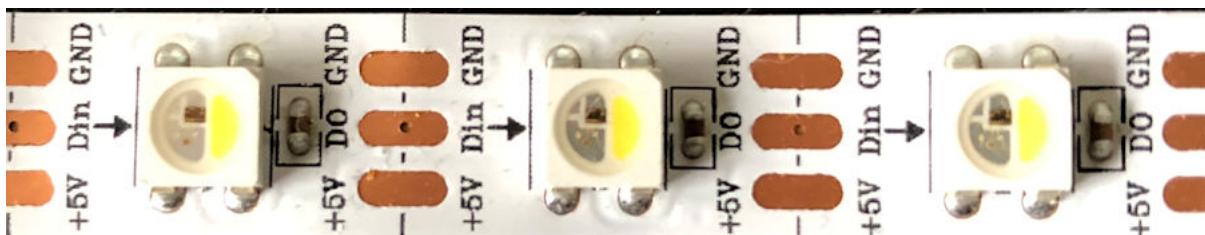
- Eventuele ondersteuning via email.

[De handleiding](#) beschrijft uitvoerig de bediening van de klok

- met de rotarydraaiknop
- of gecontroleerd via het menu met de [Bluetooth apps voor iOS en Android](#)
- of besturing met een gekoppelde PC.



Kleuren zijn in werkelijkheid veel kleuriger dan op de foto.



De woordklok bestaat uit de onderdelen zoals in de tabel hieronder is weergegeven.
Daarnaast is in de software de optie om een ATMEGA 1284 processor te gebruiken. Ik heb [printplaten hiervoor](#).

De woordklokonderdelen zijn voor 25x25cm en 50x50cm woordplaten ontworpen.
Hierbij is de hartafstand tussen de letter 1.67 of 3.33 cm. De LED-strips hebben 30 of 60 LEDS per meter op een strip.
De 25cm klok heeft 60 LEDs per meter strips en de 50cm klok 30 LEDs/meter.
In de tabel staan de prijzen waarvoor je de onderdelen, indien nog voorradig, bij mij als zelfbouwpakket kan bestellen.

De woordplaat zijn door een beletteraar met de hand gemaakt.
De bouwtijd van een klok is 20 - 30 uur.



[Frontplaat Font Thomaha NL, DE, FR, UK](#)

Benodigheden

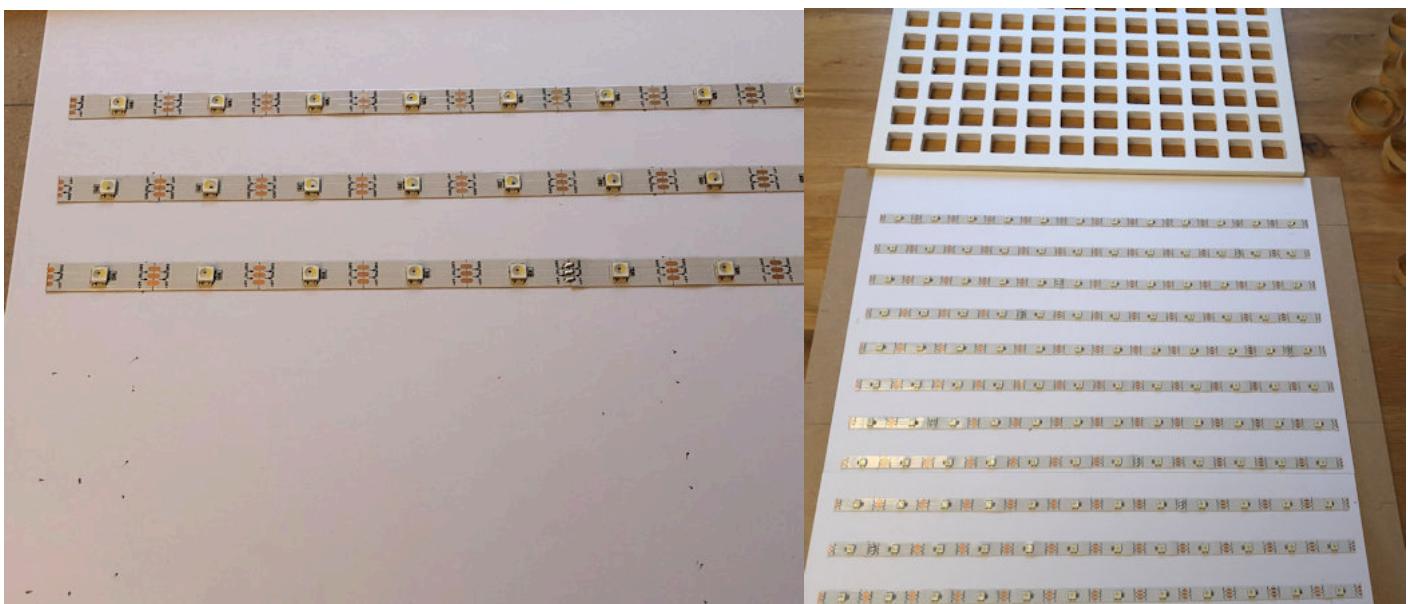
Soldeerbenodigheden
Multimeter

[Excelsheet met onderdelen](#)

| | | |
|---|----------------------------|--|
| 1 x Kast voor 25x25 cm woordplaat | Op aanvraag +/- €100,00 | |
| 1 x Woordplaat 25x25 cm met font Tomaha. Vinyl op museumglas. | +/-€70,00 | |
| 1 x Spacerplaat, geschuimd PVC wit 10 MM RAL 9003, zonder gaten | €20,00 | |
| 1 x Printplaat Fibonacci/SK6812-klok | €15,00 | |
| 1 x 1000 µF condensator | €0,30 | |
| 1 x 1.1kΩ weerstand | €0,15 | |
| 1 x 2.2kΩ weerstand | €0,15 | |
| 1 x 470kΩ weerstand (1-3 nodig) | €0,15 | |
| 1 x LED rood | €0,15 | |
| 1 x LED geel | €0,15 | |
| 1 x 2-pin female connector | €0,25 | |
| 1 x 3-pin female connector | €0,25 | |
| 1 x 5-pin female connector | €0,25 | |
| 2 x 6-pin female connector | €0,50 | |
| 2 x 15-pin female connector | €0,75 | |
| 1 x Arduino Nano / Every | €20,00 | |

| | | |
|--|---------|---|
| 1 x KY-040 Keyes Rotary Encoder | €2,00 |  |
| 1 x 4x3 membrane keypad | €10,00 | |
| 1 x RCT DS3231 Precisieklok module ZS-042 | €5,00 |  |
| 1 x CR 2032 3V lithium batterij | €3,00 |  |
| 1 x lichtsensor | €0,85 |  |
| 1 x 22kΩ weerstand | €0,15 |  |
| 1 x Adapter 5V gelijkstroom, 2 Ampere | €12,50 |  |
| 1 x Female 5.5 x 2.1mm DC Power plug | €2,00 | |
| 1 x Verlichtingsgrondplaat 3 x 300 x 300 mm MDF-plaat | €3,00 | |
| 1 x Adapter 5V gelijkstroom, >=2 Ampere | €20,00 |  |
| 100 Leds (60 LEDs/m) SK6812 Full color led strip | €40,00 |  |
| Wireless Serial 6 Pin Bluetooth RF Transceiver Module HM10 (voor Iphone, IPad) | €10,00 |  |
| Wireless Serial 6 Pin Bluetooth RF Transceiver Module HC05 (Android, W10) | €10,00 |  |
| DCF77 DCF-2 module | €20,00 |  |
| Gesoldeerde en geteste printplaat SK6812-klok | €60,00 | |
| Gesoldeerde en geteste verlichtingsplaat SK6812-klok inclusief LEDs | €150,00 | |

De verlichtingsgrondplaat in elkaar zetten



De afstand tussen de LEDs op de strip zijn geschikt om een klok van 25 x 25 cm (60 LEDs/m) of van 50 x 50 cm (30 LEDs/m) te

maken.

Je kan er voor kiezen om de 144 LEDs van de klok in 12 rijen met 12 LEDs te plakken of alleen achter de letter die moeten oplichten.

Dat laatste heeft als voordeel dat er minder LEDs nodig zijn en dat je minder gaten hoeft te boren in de spacerplaat.

Er zijn dan wel veel meer soldeerpunten die elk op zich storing kunnen opleveren.

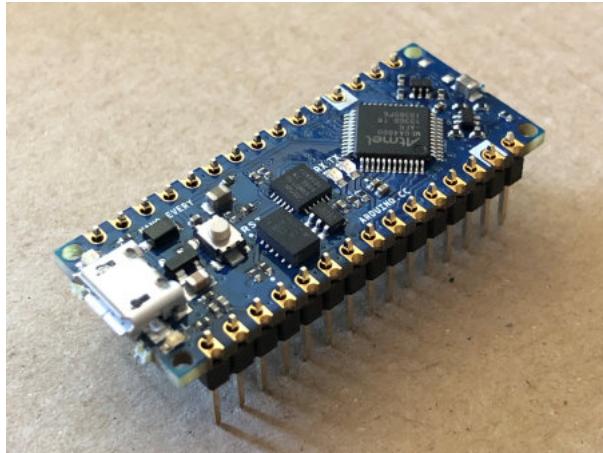
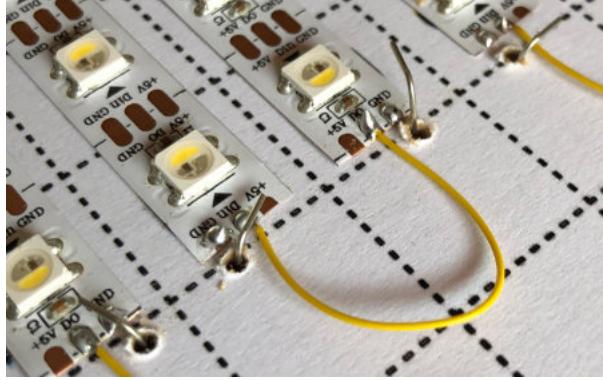
Mijn ervaring is dat niet perfecte solderingen op de strip na verloop van tijd soms losraken.

De software heeft ook een digitale tijdweergave die je niet kan gebruiken als je niet alle 144 LEDs installeert.

Je kan er ook voor kiezen om in een 1 cm dikke MDF-plaat de oplichtende woorden met een decoupeerzaag uit te zagen.

Schilder dan wel de binnenkanten helder wit anders wordt het witte licht groezelig.

Plak de strips van links naar rechts op de oneven regels en van rechts naar links over de even regels. **Volg de pijltjes op de strip.**



Ik vind het handig om vertint koperdraad te gebruiken. Je soldeert dan bijvoorbeeld alle 5V aansluitingen aan de linkerkant van de strip en alle GND aan de rechterkant van de strip
Boor hiervoor een klein gaasje door de plaat naast alle 5V aansluitingen aan de linkerkant van de strip en naar de GND aan de rechterkant
Buig een hoek van 90 graden, steek de draad door het gaatje en soldeer de draad vast aan de LED-strip.

Soldeer de pootjes van de Nano Every zo dat de lange pootjes onder zijn als de chip en USB-aansluiting boven liggen.

De nummers en teksten op de Nano moeten dan overeenkomen de de nummers op de PCB printplaat.

Breng soldeer aan op de LED-striпаansluiting en soldeer later de draad in de soldeerklopper.

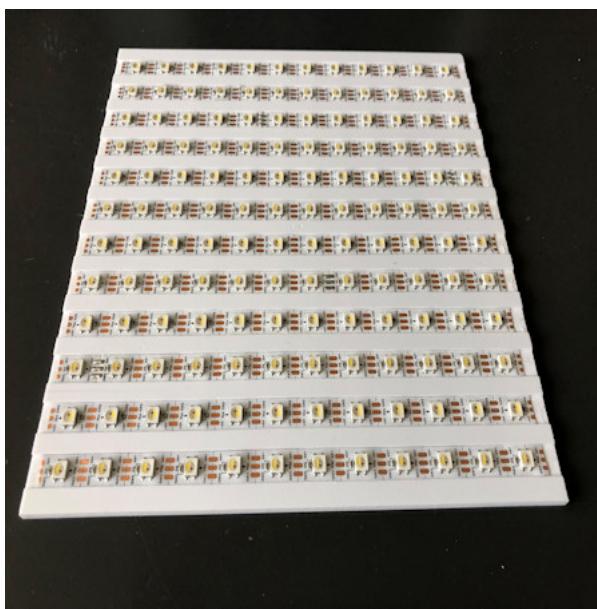
Er gaat bij volle belasting redelijk veel stroom (1A) lopen. Als je alle strips achter elkaar gaat doorverbinden loopt er bij de eerste LEDs veel stroom en moet de draad dik zijn.

Ik maak per twee regels een aansluiting vanaf de voedingsdraad met 30AWG = 0.25mm diameter. [30AWG-draad kan 0.9 A stroom verdragen](#)

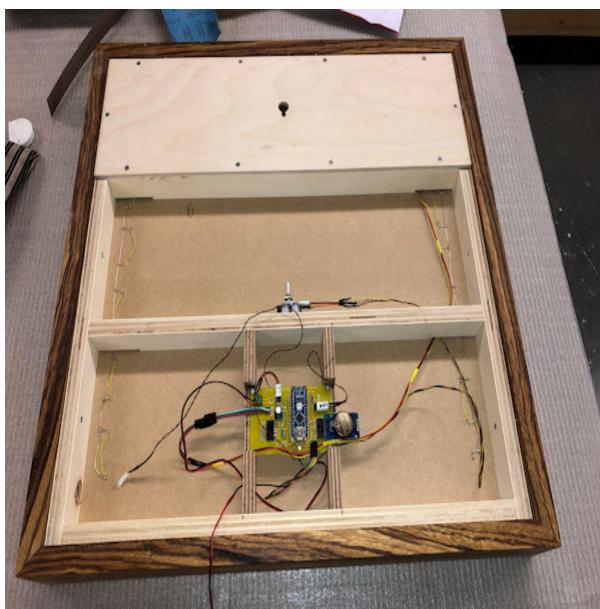
of ongeïsoleerd vertint koperdraad van 0.6mm aan de ene kant van de strips de 5V aansluiting en aan de andere kant de aarde aansluiting.

Zorg er uiteindelijk voor dat elke strip zijn voeding heeft.

Op de voorkant, op de LEDs, komt de spacerplaat te liggen en deze moet plat de ondergrond liggen om lichtlekkage te voorkomen.



Voorkant



Achterkant na bedraden

In de signaaldraad tussen LED_PIN 5 (D5) van de Arduino en de SK6812 LED-aansluiting zit een 470 Ohm weerstand. Over de GND en 5V naar de LED-strip zit een 1 mF (1000 uF) condensator om de inschakelspanning te dempen. De LEDs werken wel zonder deze twee componenten maar kunnen dan bij aanzetten van de stroom beschadigd raken. Je kan eventueel ook alle componenten zonder de printplaat direct op de pennen van de Arduino Nano aansluiten. Er is een PCB waar alles op gesoldeerd kan worden. Deze printplaat kan ook gebruikt worden voor de fibonacciklok.

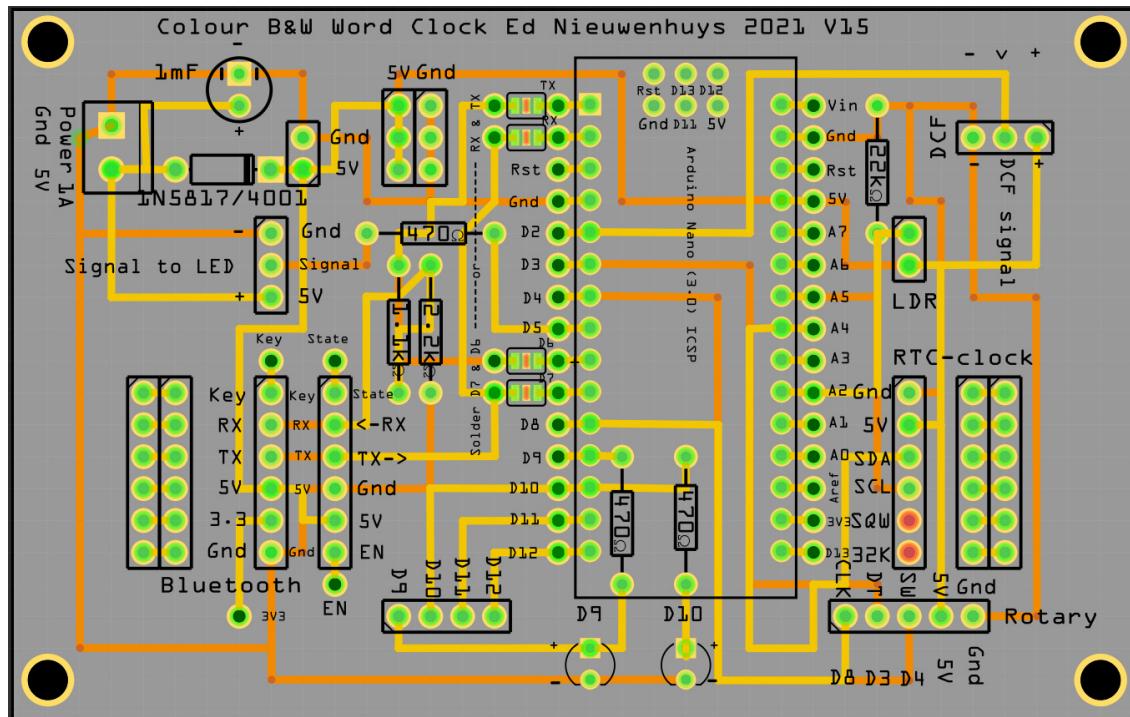
De laatste PCB [versie V15](#) heeft vaste aansluitingen voor de meest gebruikte onderdelen gekregen.

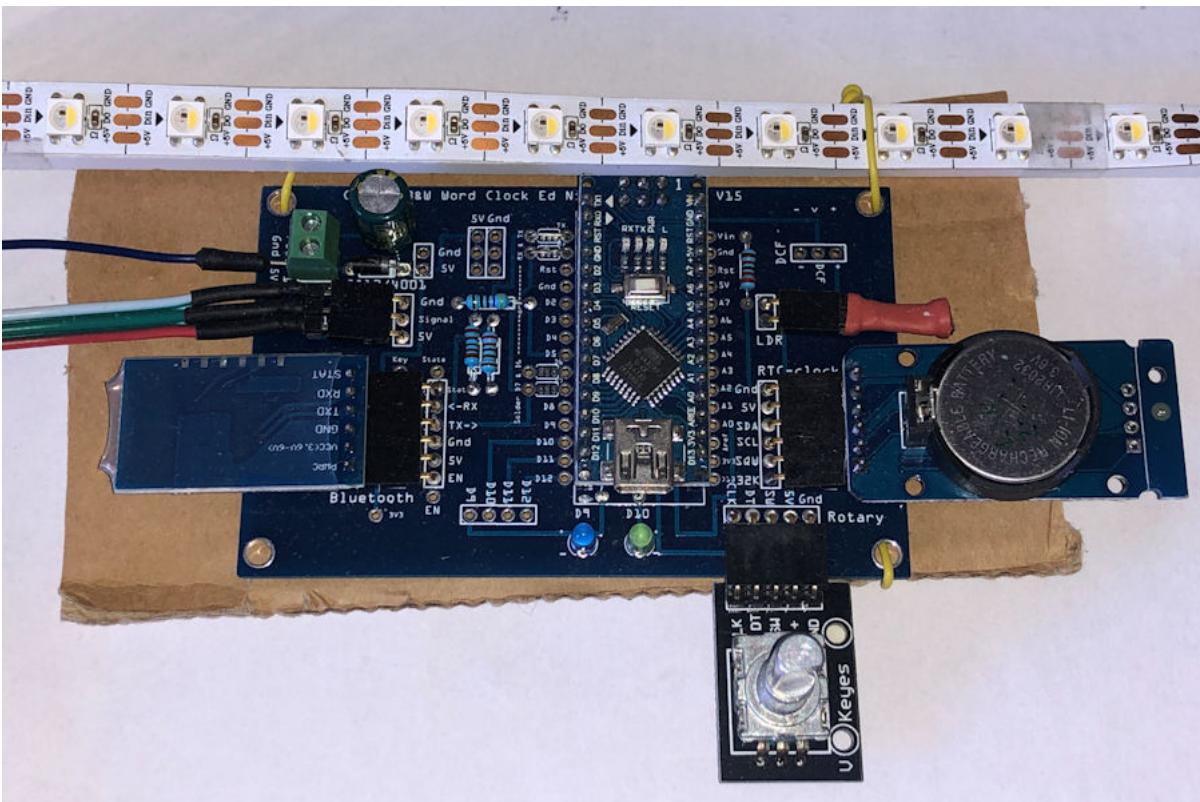
Het is nu mogelijk om te kiezen of de seriële verbinding met de Bluetooth-module via de standaard aansluitingen 0 en 1, TX en RX, gaat of dat deze via aansluitingen D6 en D7 gaat. Dan wordt een extra library in de software gebruikt die extra programmaruimte inneemt.

Voordeel is dat je het programma in de Arduino kan laden zonder de Bluetooth-module los te hoeven koppelen. Deze stoot namelijk de upload van het programma

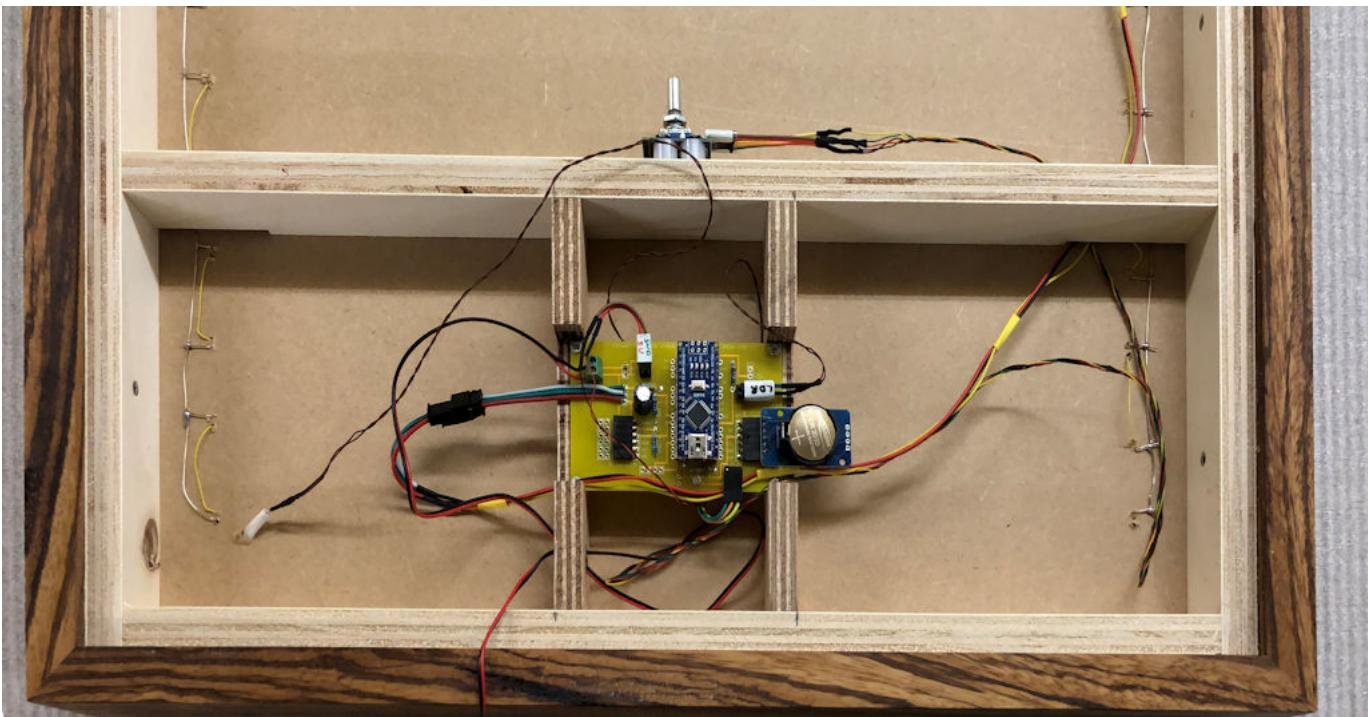
In de huidige softwareversie worden D6 en D7 gebruikt voor de Bluetooth-module en moeten de pads bij D6 en D7 dicht gesoldeerd worden.

In PCB V15 is een diode 1N5817 of 1N4001 toegevoegd om te voorkomen dat er bij het aansluiten van een lange LEDstrip er te veel stroom via de USB-poort van de Arduino Nano en aangesloten PC geleverd gaat worden. Tot 30 LEDs, en mogelijk 100 LEDs, kan de USB-poort aan stroom leveren en kan de diode vervangen worden door een draad. Maar er zit een beperking aan en en bij een te hoge stroomvraag raakt de Arduino defect





De onderdelen op zijn plek. In dit voorbeeld zijn pin 0 & 1 (Tx&RX) aan de Bluetooth verbonden



De aansluitingen in de kast

Hieronder een opsomming van de aansluitingen op de Arduino

```
// digitale poortaansluitingen D2 - D13
DCF_PIN      = 2,      // DCFPulse on interrupt pin connected to DCF77 module
encoderPinA  = 3,      // Rotary right (labeled DT on decoder)on interrupt pin
clearButton  = 4,      // Rotary switch (labeled SW on decoder)
LED_PIN       = 5,      // Pin to control colour 2811/2812 leds
BT_RX        = 6,      // Bluetooth RX, Connect to Bluetooth TX
BT_TX        = 7,      // Bluetooth TX, Connect to Bluetooth RX
encoderPinB  = 8,      // Rotary left (labeled CLK on decoder)no interrupt pin
EmptyD09     = 9,      // EmptyD09
DCF_LED_Pin = 10,     // pin for LED to display pulse received from DCF77 module
PWMpin       = 11,     // pin for LED to display intensity read from LDR
EmptyD10     = 10,     // EmptyD10
EmptyD11     = 11,     // EmptyD11
EmptyD12     = 12,     // EmptyD12
secondsPin   = 13,
```

```
// digitale poortaansluitingen A0 -A5
PhotoCellPin = 2, // LDR pin
EmptyA3 = 3, // EmptyA3
SDA_pin = 4, // SDA pin van de klok DS3231
SCL_pin = 5}; // SCL pin van de klok DS3231
```

De klok in elkaar zetten

Het is handig om de onderdelen van de klok met papierafplakband vast te plakken. Dit plakband komt makkelijk los.
Plak de verlichtingsplaat en spacer aan elkaar vast.

Plak over de spacer een wit vel papier. Papier geeft een mooie tekening aan de verlichte letters.

Leg de woordplaat over het papier en fixeer deze ook met plakband.

Bij een hardhouten klok wordt, afhankelijk van de uitvoering, de woordplaat in de sleuf in de kast geschoven.

Monter daarna de vierde, onderkant van de kast.

Verlijm dit of plak het met plakbank tijdelijk vast.

Als alles goed gelabeld is kunnen de klok, rotary encoder, LDR, voeding aangesloten worden.

Monter de kast af.

Bluetooth-verbinding

Met een HC05 of HM-10 of JDY-23 Bluetooth-module kan er een Bluetooth-verbinding met de klok gemaakt worden.

De HC05 communiceert alleen met Android. De HM-10 en JDY-23 communiceert zowel met Android als Apple-IOS.

In onderstaande link staat beschreven aan welke pins de module is aangesloten en hoe de naam van de module aangepast wordt.

Met een [Bluetooth-terminal app](#) op de telefoon kan de tijd als hhmmss of hhmm naar de klok gestuurd worden.

De klok stuurt elke minuut gegevens, naar de terminal app terug.

Voor Windows phone, Android en Iphone zijn Bluetooth terminal apps te vinden.

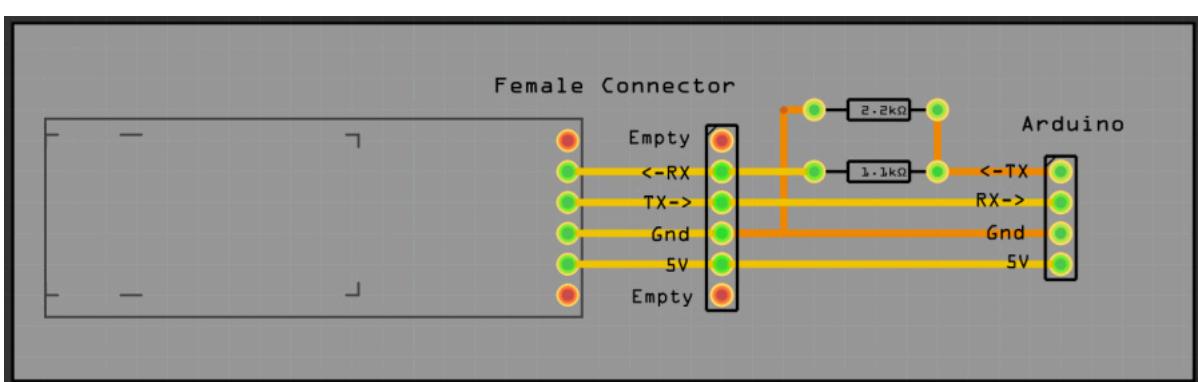
De Bluetooth terminalprogramma's zijn ook voor PC's te vinden.

De Bluetooth-module communiceert tussen RX (read) en TX (transmit) met 3.3V.

Het is mogelijk om hem direct op de Arduino-poorten aan te sluiten maar het verlengt de levensduur van de module als de spanning naar 3.3V wordt verlaagd.

Dit kan met een 3.3V-5V TTL Level Logic Level Converter module, of met een "voltage divider"-schakeling met weerstanden zoals hieronder weergegeven.

Je kan de weerstanden ook in de bedrading van de module naar de printplaat monteren.



Bluetooth met de HM10 BLE voor Apple IOS

Met deze HM10 BLE module kan gecommuniceerd worden met Apple IOS en Android.

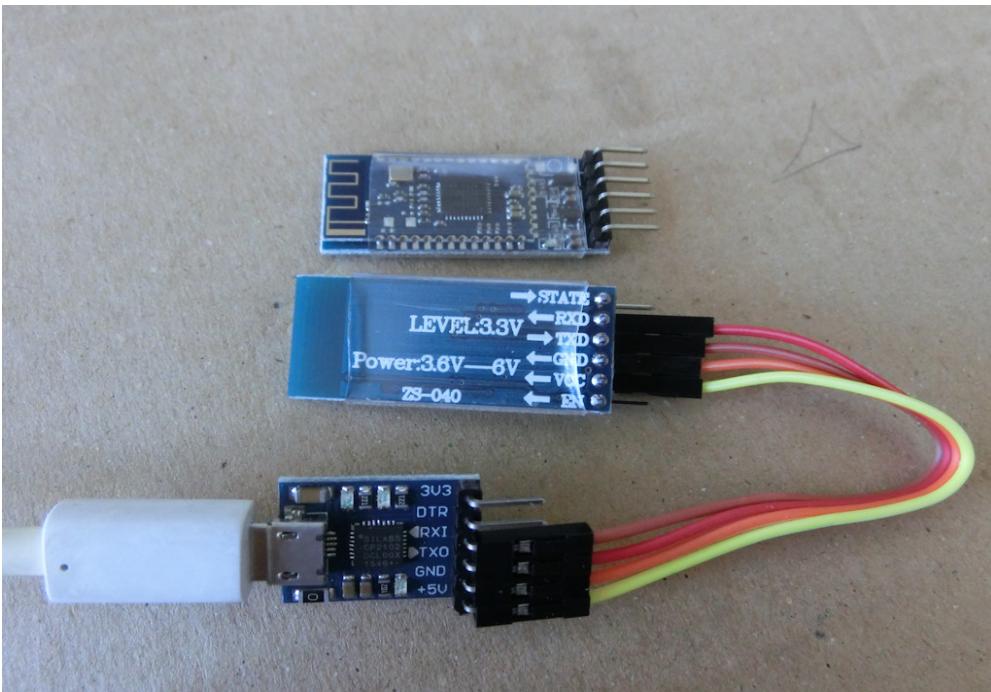
Daarna kan met een serial terminal programma ([BLEserial HM-10 voor IOS](#) of voor Android BLE scanner van Blue Pixel Technologies) commando's worden overgestuurd en de klokuitvoer opgevangen worden.

Gekoppeld via een FTDI kan met de Arduino serial monitor met de Bluetooth-module AT-commando's gegeven worden.

Zie vorige paragraaf voor AT commando's

AT+NAMEnieuwenaam geeft de module de naam: nieuwenaam

[Meer info over de Bluetooth-communicatie hier.](#)



DCF77-ontvangst met DCF-2 module



Helaas storen de IC's in de kleuren-LED's de DCF-ontvangst.

Met deze kleuren-LEDs moet de DCF ontvangstmodule, zeker 10 cm of meer van de LEDs opgehangen worden en is een goede storingvrije voeding essentieel.

Met een LED op de PCB of met de seriële communicatie is de ontvangst te volgen.

Deze DCF77-module heeft drie aansluitingen; +, - en signal.

Sluit de + aan op 5V en de - op GND.

Signal gaat naar pin 2.

NB Pin 2 werd gebruikt voor de rotary encoderPinB left (labeled CLK on decoder). Deze wordt nu aangesloten op P8. (Pin 2 is een interrupt pin. Voor een optimale DCF-ontvangst is het reageren op een interrupt niet voor de rotary encoder noodzakelijk.)

Kijk in de source van de versie van de software onder "PIN assignments" voor de juiste aansluiting.

[Hier achtergrondinformatie over DCF.](#)

Energieverbruik klok

De klok verbruikt bij 5V als de LEDs op halve sterkte branden 0.15A.

Als alle LEDs branden, is het verbruik 0.6 A.

Bij normaal gebruik is een voeding van 5V 1A voldoende.

Source code

Op deze [Github link](#) zijn deze en nieuwe versies ook te vinden.

[Character_Colour_Clock_V076](#)

[Handleiding_SK6812-WS2812-woordklok V076.pdf](#)

Dit is een stabiele versie die meer dan een jaar in klokken draait

Laatste versie V087 + benodigde libraries: [Character_Colour_Clock_V087.zip](#)

[Handleiding_SK6812-WS2812-woordklok V087.pdf](#)

De zip file bevat twee INO bestanden V087 en V087tiny

V087 bevat tevens nog de code voor de 4-talenklok, is zeer uitgebreid en ondersteund diverse displays en invoermiddelen.

V087tiny bevat alleen code voor de Nederlandse woordplaat met een rotary draaiknop en Bluetooth.

Deze uitvoering is voor de meeste toepassingen voldoende.

Software op [Github](#)

Software

De gebruikte libraries moeten geïnstalleerd zijn en zijn allemaal in de Arduino IDE te vinden.
Voor de zekerheid zijn in een ZIP-file de gebruikte libraries verpakt

```
#include "RTClib.h"          // https://github.com/adafruit/RTClib
#include <TimeLib.h>           // For time management
#include <Adafruit_NeoPixel.h>   // https://github.com/adafruit/Adafruit_NeoPixel
#include <SoftwareSerial.h>      // For Bluetooth
#include <Encoder.h>            // http://www.pjrc.com/teensy/td_libs_Encoder.html
#include "DCF77.h"              // http://playground.arduino.cc/Code/DCF77
```

Bovenin de source wordt met #defines aangegeven welke code voor de Arduino Nano Every gecompileerd moet worden.
Met // voor een #define zet je de source code voor dat onderdeel uit.

Bijvoorbeeld:

```
-----  
// Definition of installed modules  
-----  
  
// -----> Define only one library  
#define NEOPIXEL          // Adafruit Neopixel for WS2812 or SK6812 LEDs  
//#define LIBSK6812          // SK6812 library. Only with SK6812 LEDs (saves 614 bytes with NEOPIXEL)  
  
// -----> Define which module is present.  
#define BLUETOOTHMOD        // Use this define if Bluetooth needs other pins than pin 0 and pin 1.  
//#define BLEnRF52MOD         // turn on for RP2040, Nano BLE33, MKR1010 etc  
#define MOD_DS3231           // The DS3231 module is installed, if not the Arduino internal clock is used  
//#define LCDMOD              // For LCD support  
#define ROTARYMOD            // Rotary encoder installed
```

In dit voorbeeld wordt de Adafruit Neopixel library gebruikt en de library LIBSK6812 niet.

```
#define NEOPIXEL  
//#define LIBSK6812
```

Volgend voorbeeld:

```
#define BLUETOOTHMOD        // Use this define if Bluetooth needs other pins than pin 0 and pin 1.  
//#define BLEnRF52MOD
```

De Bluetooth-module gebruiken wij maar is niet aangesloten op pin 0 en 1 maar op pin 6&7.

Om pin 6&7 te sparen kan de de Bluetooth-module ook op de seriële poort pinnen 0 en 1 aangesloten worden.

Maar als een programma in de Nano Every wordt ge-upload **moet de Bluetooth-module los!**

Het programma wordt dan namelijk ook in de Bluetooth-module gestuurd die hevig gaat protesteren.

Kijk verder in het programma hoe pin 6&7 worden gebruikt met de library SoftwareSerial.

In de source code vind je code tussen #ifdef en #endif.

Deze code wordt dan alleen gecompileerd als dat onderdeel gedefinieerd is met een #define.

Het gebruik van deze definities scheelt programmaruimte maar ook onnodige opdrachten van een module dat niet aanwezig is maar wel door de software gelezen wordt op niet aangesloten pinnen. Dat kan alleen maar gaan storen.

Als er ruimtegebrek ontstaat kan je mijn library SK6812 gebruiken. Die library gebruikt iets minder geheugen.

NL144NL is om de 144 LED display met Nederlandse tekst te selecteren. FOURLANGUAGECLOCK is om een vier-talenklok te maken.

De overige defines zijn om de diverse modules te selecteren.

MOD_DS3231 is de tijdsensor.

Om te testen kan het handig zijn om deze uit te zetten. Dan wordt de interne Arduinoklok gebruikt.

Deze is veel onnauwkeurige en wijkt seconden per uur af terwijl de DS3231 maximaal 5 seconden per miljoen seconden afwijkt.

Als de klok dus niet gelijk loopt controleer dan of de juiste optie is gekozen.

Bijvoorbeeld:

```
#ifdef MOD_DS3231
RTC_DS3231 RTCklok;
#else
RTC_Millis RTCklok;
#endif
```

Er zijn DS3231 modules in omloop die erg afwijken. Dan zal je deze moeten vervangen.

De DCF77 ontvanger is voor de hobbyist. Soms lukt het dagen niet om de tijd te ontvangen.

Dat kan liggen aan een slechte voeding aan de klok of op hetzelfde stopcontact.

Soms hebben de buren een stoerbron aan maar ook de kleuren LEDs in de klok storen.

Soms zijn het gewoon slechte ontvangstomstandigheden.

Ik heb daarom een grote [DCF-ontvanger gemaakt](#) en [een kleine](#), waarvan ik weet dat deze het doet.

Als mijn nieuwe ontwerp het naast deze ontvanger ook niet werkt weet ik dat het aan de software waarschijnlijk ligt.

De ontvanger moet in ieder geval minimaal 15 cm van de klok liggen met de ferrietstaaf ongeveer Noord-Zuid gericht
Dus dwars op de zender in Duitsland. De gele LED moet regelmatig knipperen.

Met de menu-optie A kan het signaal onderzocht worden.

Meer [DCFNanoEvery/DCFHC12Transmitter](#) en hier [DCF77 transceiver/DCFtransceiverklok](#).

```
// ======  
//
```

```

//-----
// ARDUINO Definition of installed modules
//-----
// -----> Define only one type of LED strip
// #define LED2812          // Choose between LED type RGB=LED2812 or RGBW=LED6812
#define LED6812           // Choose between LED type RGB=LED2812 or RGBW=LED6812
// -----> Define only one library
#define NEOPIXEL           // Adafruit Neopixel for WS2812 or SK6812 LEDs
// #define LIBSK6812          // SK6812 library. Only with SK6812 LEDs (saves 614 bytes with NEO
// #define FOURLANGUAGECLOCK // Selectt this define for the 4-language clock with 625 LEDs
// Or select this define for the Dutch clock with 144 LEDs4
#define NL144CLOCK

// -----> Define which module is present.
#define BLUETOOTHMOD        // Use this define if Bluetooth needs other pins than pin 0 and pin 1.
// #define BLEnRF52MOD         // turn on for RP2040, Nano BLE33 , MKR1010 etc
// #define WIFIMOD             // When WIFI module is available
// #define HC12MOD              // Use HC12 time transceiver Long Range Wireless Communication Mo
#define DCF77MOD             // DCF77 receiver installed
#define MOD_DS3231            // The DS3231 module is installed, if not the Arduino internal clock is used
// #define LCDMOD               // For LCD support
// #define HT16K33tijd          // 4-digit HT16K33 time display installed https://www.adafruit.com/
#define ROTARYMOD             // Rotary encoder installed
// #define KEYPAD3x4            // Use a 3x4 keypad with 7 wires
// #define KEYPAD3x1            // Use a 3x1 keypad with 4 wires
// #define ONEWIREKEYPAD3x1      // Use a 3x1 keypad with one wire
// #define ONEWIREKEYPAD3x4      // Use a 3x4 keypad with one wire

#define HET     ColorLeds("Het",      0,   2,  MINColor);
#define IS      ColorLeds("is",       4,   5,  SECCColor); ColorLeds("", 8,10, 0); Is = true;
#define WAS    ColorLeds("was",      8,   10, SECCColor); ColorLeds("", 4, 5, 0); Is = false;
#define PRECIES ColorLeds("precies", 17,  23, LetterColor);
#define MTIEN   ColorLeds("tien",    12,  15, LetterColor);

```

Met `#define` is het ook mogelijk om meerdere opdrachten samen te voegen tot een tekst.

Wat in bovenstaande `#define` belangrijk is de functie `ColorLeds()`.

Deze geeft aan welke LEDs moeten oplichten bij een woord.

De eerste LED heeft nummer 0 (nul)

WAS wordt door de LEDs 8, 9 en 10 verlicht.

De nummering kan aangepast worden als het ontwerp iets anders is geworden.

Bijvoorbeeld als vergeten is de LEDs elke regel van richting te verander bij het oppakken. (-:

Na allerlei initialisaties die tussen `#ifdef` en `#endif` gebruikt worden volgt de `Setup()` functie.

Hierin worden de gekozen modules gestart.

De `setup()` wordt één keer doorlopen bij de start van de Arduino.

Daarna gaat het programma naar de `loop()` functie.

Het programma blijft hier in rondlopen en komt we er alleen uit met de functies die hierin worden aangeroepen.

In dit programma :

```

Void loop()
{
    InputDevicesCheck();
    if (Demo)
        Demomode();
    else if (Zelftest)
        Selftest();
    else
        EverySecondCheck();
}

```

`InputDevicesCheck()` controleert of er invoer van iets is en verwerkt dat.

Bijvoorbeeld `SerialCheck()`; `RotaryEncoderCheck()` en `BluetoothCheck()`;

Verder is de `EverySecondCheck();` belangrijk. Daar gebeurt van alles dat binnen de seconde moet gebeuren

Daarna volgen `EveryMinuteUpdate`, `EveryHourUpdate`, `EveryDayUpdate`.

Persoonlijk vind ik dit een mooie methode om niet onnodig veel tijd in functies te verspillen.

Bijvoorbeeld het dimmen van de display als je een vinger voor de LDR houdt kan best één keer per seconde gecontroleerd worden.

De tijd verzetten hoeft maar een keer per minuut. Et cetera

Invoer van de seriële poort of Bluetooth wordt verwerkt in `ReworkInputStream()`;

Dit waren de basics van de programmaflow.

Bij verdere onduidelijkheden kan je mij mailen.

<- [Terug naar startpagina](#)

[Bestellijst in Excel](#)

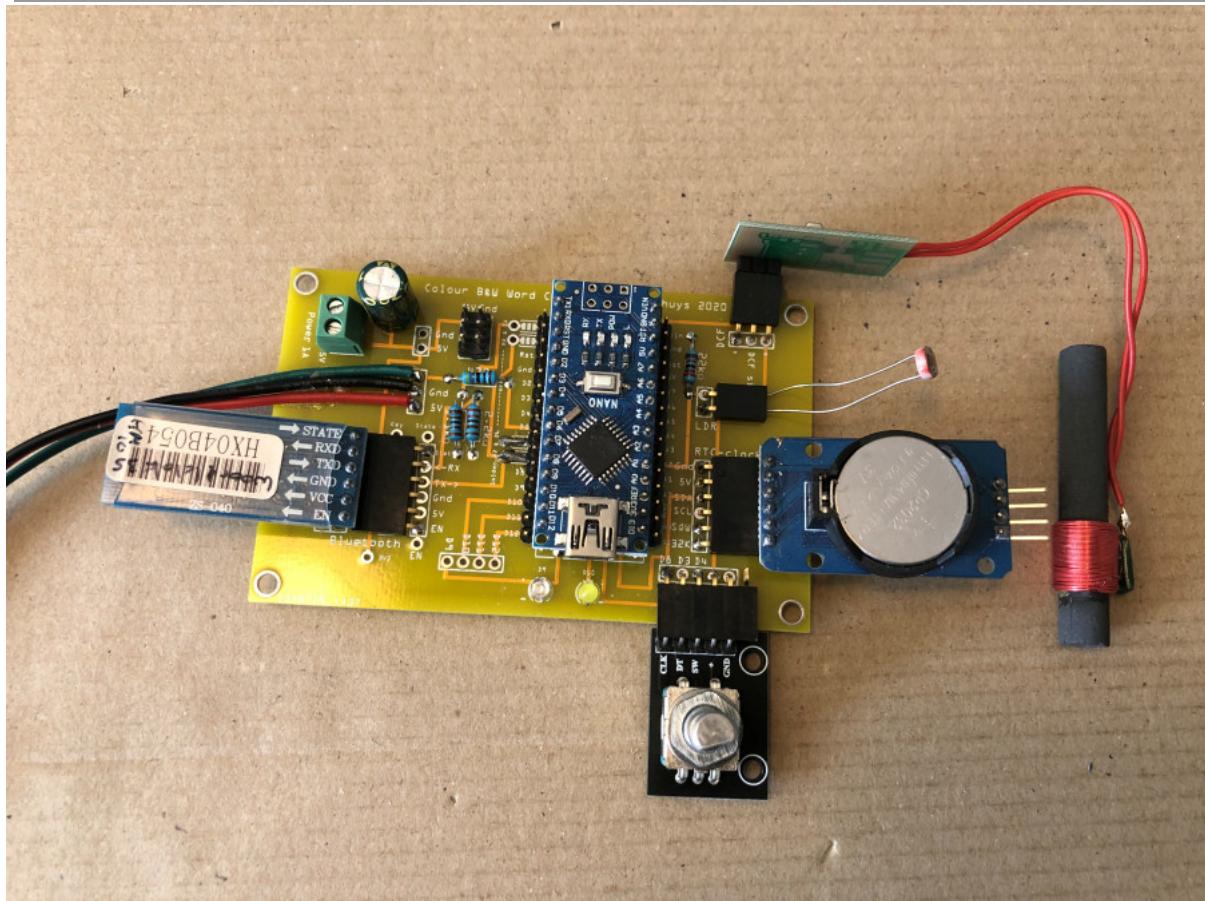
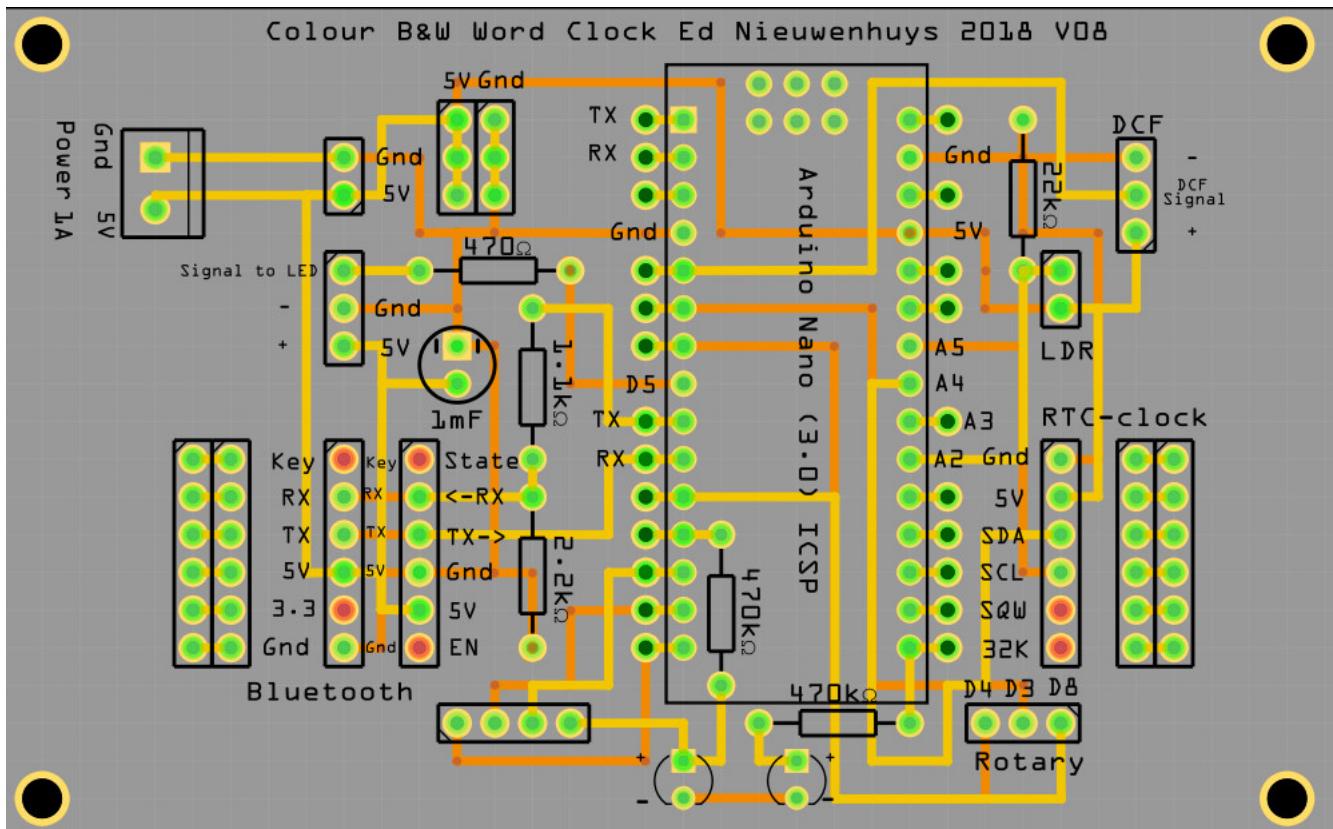
Deze woordklok met een ESP32 wordt beschreven [hier op Github](#).

[Email](#)

Ed Nieuwenhuys. 22 augustus 2024

16 april 2023, 18 okt 2022, 27 mei 2021, 18 maart 2021, 13 Nov 2020, 5 augustus 2020, 23 december 2019

Hieronder de oudere PCB versie 08.



<- [Startpagina](#)

Oud

Handleiding voor Arduino Nano Every met SK6812 of WS2812 LEDs

[Handleiding_SK6812-WS2812-woordklok_V069V070.docx](#)

[WoordklokSK6812_files/Character_Colour_Clock_V070.zip](#)

Voor 144 LEDs met Arduino Nano Every

Sinds 2020 de Arduino Nano Every met 48Kb geheugen ipv 32Kb zeer goedkoop officieel aan te schaffen

In deze versie kan een DCF77-ontvanger gebruikt worden.

Er zijn ook opties aanwezig om met de MKR1010 met WIFI te werken of te compileren voor de ATMEGA 1280/1284 chips.

Versie voor Arduino Nano Every met 144 LEDS.

[Character_Colour_Clock_SK6812_1284-V036-Nano-144_LEDs.ino](#)

Voor maximaal 96 LEDs met Arduino Nano of Arduino mini

De source code om de klok aan te sturen is te gebruiken voor de Arduino Nano Uno en mini tot versie V076.

Dan moet wel een klok met maximaal 96 LEDs gemaakt worden.

Dat is 1 LED onder elke letter die gebruikt wordt om de tijd weer te geven.

Houd de **programmagrootte onder 23572 bytes!**

De LED-libraries gebruikt geheugen dat niet afgeschreven wordt. Als het programma te groot wordt gaat bijvoorbeeld de seriële en/of bluetooth communicatie ineens niet meer.