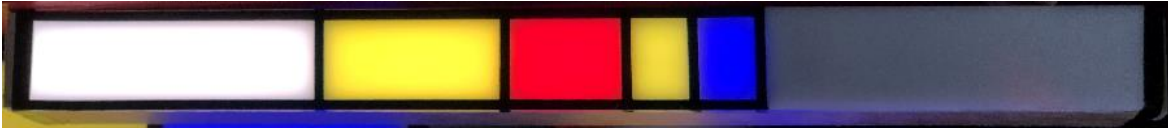
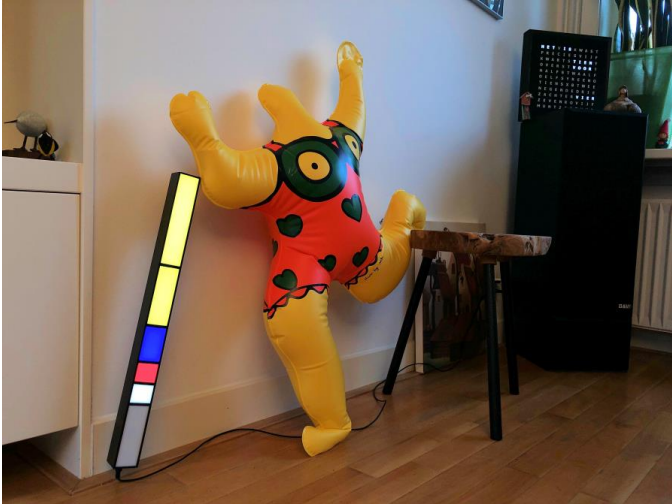


Fibonacci-vierkante kokerklok



Deze vierkante kokerklok is een variatie op de [kast Fibonacci klok](#). De software is vereenvoudigd en geschikt voor twee soorten kleuren-LEDs.



De Fibonacci-klok was een kick-starterproject van [Philippe Chrétien](#). Wat deze klok zo aantrekkelijk maakt is zijn eenvoud, de vindingrijkheid en de voortdurend veranderende Mondriaan-gelijkende schilderijendisplay. Mondriaan was ook op zoek naar eenvoud in zijn schilderijen dat begon bij realisme en via kubisme bij zijn kermerkende stijl eindigde. Naast het kleurspel nodigt deze klok uit tot het narekenen van de gepresenteerde tijd. Maar een klok? En wat heeft Fibonacci met

Mondriaan te maken?

De gekleurde vlakken gescheiden door zwarte lijnen doen meteen denken aan de schilderijen van [Piet Mondriaan](#) als de primaire kleuren rood, geel en blauw worden gebruikt.

In deze klok wordt gebruik gemaakt van kleuren en oppervlakten die de reeks van [Fibonacci](#) volgen.
Rood + Blauw = uur, (Geel + Blauw) x 5 = minuten.
 $5 + 3 = 8$ uur, $1 + 3 = 4$, $4 \times 5 = 20$ minuten Het is 8:20, tien voor half negen.

Wat de klok doet is de verlichting aanzetten achter de vlakken die opgeteld moeten gaan worden.

Om de uren en minuten te combineren wordt een derde kleur, blauw, gebruikt.

De rode vlakken zijn de uren.

De gele vlakken vermenigvuldigd met 5 zijn de minuten en de blauwe vlakken als er een rode en gele hetzelfde vlak bezetten.

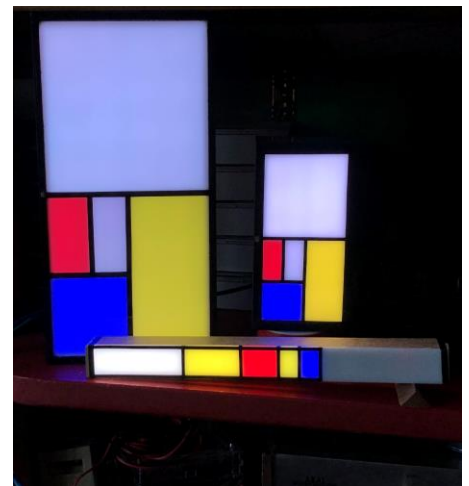
De witte vlakken betekenen nul, die stellen niets voor met optellen.

Fibonacci was een Italiaanse wiskundige die de naar hem genoemde getallenreeks ontwikkelde. Het was niet een normale getallenreeks maar een die ook vaak in natuurlijke processen voorkomt. Bijvoorbeeld in zonnebloemen en schelpen.

De reeks is simpel. Elk volgend getal in de reeks is de som van de twee voorgaande: 1, 1, 2, 3, 5, 8,

Als wij 1, 1, 2, 3, 5 optellen komen wij op 12.

Ah, precies het aantal uren in een dagdeel en 12 vermenigvuldigt met 5 is het aantal minuten in een uur.



Fibonacci-vierkante kokerklok

Voorbeeld: 4 uur.

Dat kan een optelsom zijn van : $1+1+2$ of $1+3$.

Voorbeeld: 8 uur.

Dat kan bereikt worden met het optellen van $1+1+2+4$, of $1+3+4$ of $3+5$.

De display van de klok is een weergave van de Fibonacci-reeks en de oppervlakte van de vlakken is de verhouding van de oppervlakte in de reeks.

Als het nu duizelt is het goed.

Nog een voorbeeld. 3:35.

Dat zijn drie rode (uren) vlakken en $35/5 = 7$ gele (minuten) vlakken.

Drie kan je maken met het 1 vlak + 2 vlak of alleen met het 3 vlak.

Zeven kan zijn $1 + 1 + 5$ of $2 + 5$ of $1 + 1 + 2 + 3$.

3:35 tot en met 3:39 geven dezelfde 7 minutenvlakken maar de kloksoftware probeert elke minuut een andere combinatie te vinden. Dit gaat random dus het kan zijn dat hij weer dezelfde combinatie vindt en niet verspringt.

De bouw van de kast

De minimale ruimte die nodig is om de elektronica in een kast te bouwen is $25 \times 25 \times 100$ mm binnenmaten als er een Arduino Nano wordt gebruikt waarvan de aansluitpennen aan een kant zitten.

Maak anders de kast wat hoger en eventueel breder om de zijkant van de kast vierkant te houden

De strip heeft 12 LEDs per 20 cm (60 LEDs per meter). De kleinste kast wordt daardoor 30 cm lang. Bij een kant-en-klare Arduino, zoals hieronder achterin op de foto, zitten de aansluitpennen aan de boven- en onderkant. Dit maakt de kast 15 mm hoger.

Inclusief de aansluitdraden is dan een kast van 40 mm breed of hoog nodig.

De eind 2019 uitgekomen [Nano Every](#) heeft de zes pennen niet meer is daardoor lager (en ook nog goedkoper).

De binnenkant van de compartimenten waar de verlichting in brandt moet perfect wit zijn. Je kan daar bijvoorbeeld MDF voor gebruiken dat wit is.

Maar het kan ook eenvoudig door wit papier aan de binnenkant van de compartimenten plakken.

Als scheiding tussen de compartimenten kan dun stevig karton, MDF of ander materiaal gebruikt worden. Maak het niet dikker dan 2 mm.

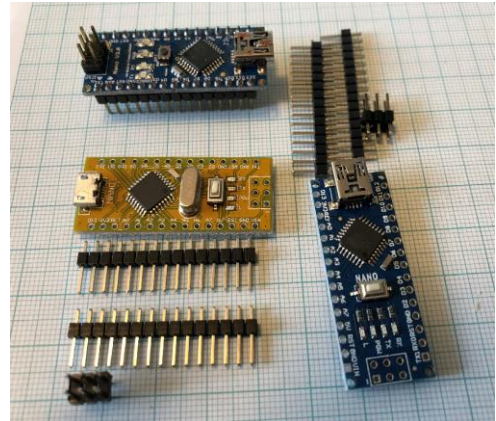
Het is mooi als de witte perspex bovenplaat in het hout valt.

Zaag hiervoor met een zaagtafel een sleuf van 1 mm diep in de zijkant, 1-2 mm onder de bovenkant ter dikte van de perspexplaat.

Maar het kan natuurlijk ook tussen de zijkanten worden geplakt.

Voor de kleinste kast met een LED-strip van 60 LEDs per meter:

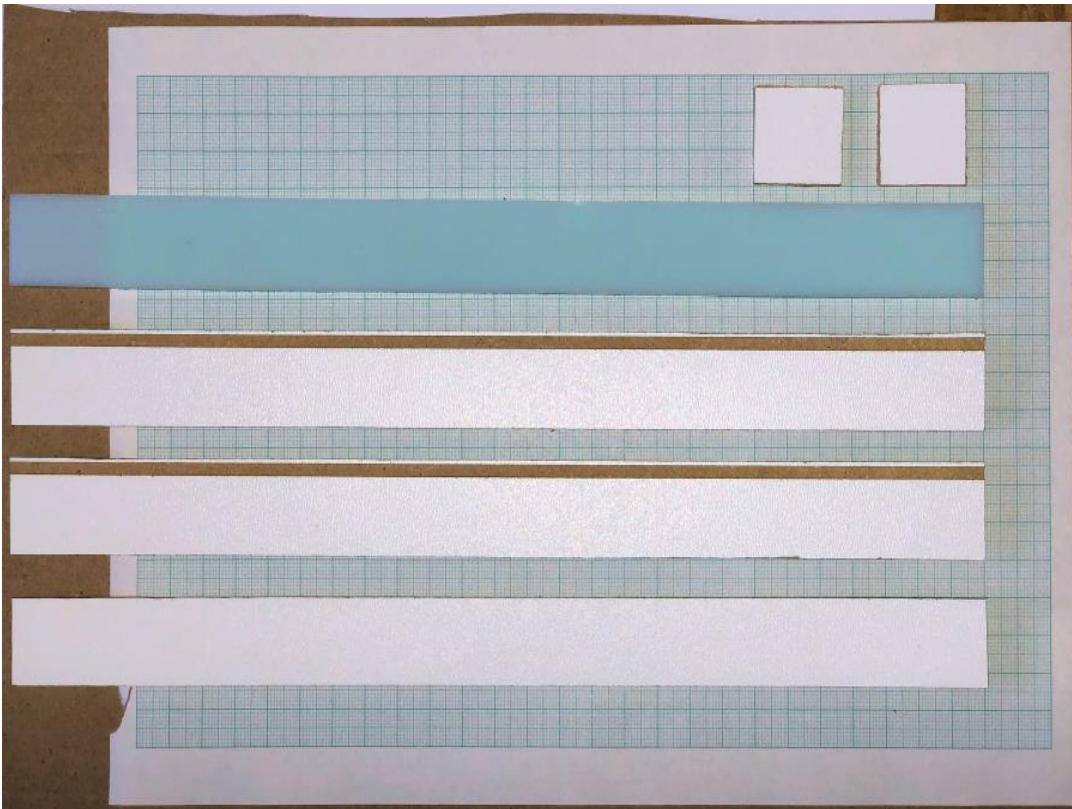
- Zaag een strook voor de bodem 25 mm breed en 300 mm lang.
- Zaag meteen een strook van 25 mm breed (maximaal 2 mm dik) voor de afscheidingen voor de compartimenten en zaag de zeven afscheidingen op maat (25×25 mm).



Fibonacci-vierkante kokerklok

Als de compartimentafscheidingsen uit karton worden geknipt kan dat later.

- Zaag twee zijkanten van 35 cm op een breedte van 25 mm + de dikte van de bodemplaat.
- Zaag eventueel een sleuf net onder de bovenrand uit waarin het perspex kan vallen.
- Zaag de zijkanten van de kopse kant uit met een breedte van 25 mm.
- Plak wit papier op alle tussenschotten en zijkanten van de kast waar de verlichting komt.
- Plak de kast in elkaar op een kopse kant na.
Handig! Gebruik papieren afplakband om de onderdelen stevig op elkaar geklemd te houden.
- Plak de LED-strip, van de gelijmde kopse kant af, op de bodem vast.
- Breng aan de zijkanten van de strip een druppel Bison-kit aan om te voorkomen dat de zijkant van de strip in de loop der jaren opkrult.
- Breng de elektronica aan (zie hieronder) en lijm de tussenschotten vast.
- Werk de kast netje af als de elektronica naar behoren werkt
- Plak met 2 mm breed zwart plakband of teken met een zwarte viltstift de compartimentscheidingen af op het perspex voor een beter contrast tussen de lichtvlakken.



Voor de grote 52 cm kast met een LED-strip van 30 LEDs per meter:

Voordeel van de grotere kast is dat dan een grotere en nauwkeuriger DS3231-tijdsmodule gebruikt kan worden.

De maten zijn dan 40 x 40 mm per compartiment en de lengte van de kast wordt 520 mm.

De LED-strip bevat dan 30 LEDs per meter.

Fibonacci-vierkante kokerklok

De elektronica in elkaar zetten

De benodigdheden voor de elektronica is hieronder weergegeven

Een soldeerbout is nodig om drie draden aan de LED-strip te solderen.

De lichtgevoelige sensor en de weerstand zouden bij voorkeur ook gesoldeerd moeten worden.

Een multimeter van 10 euro is handig om de weerstanden te kunnen meten en eventuele slechte verbindingen te controleren.

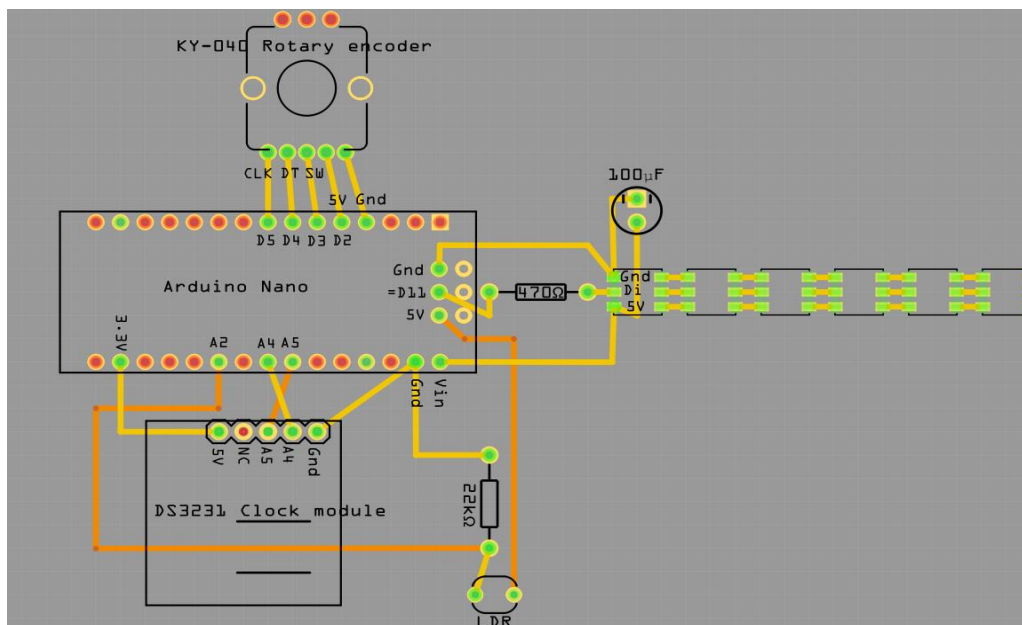
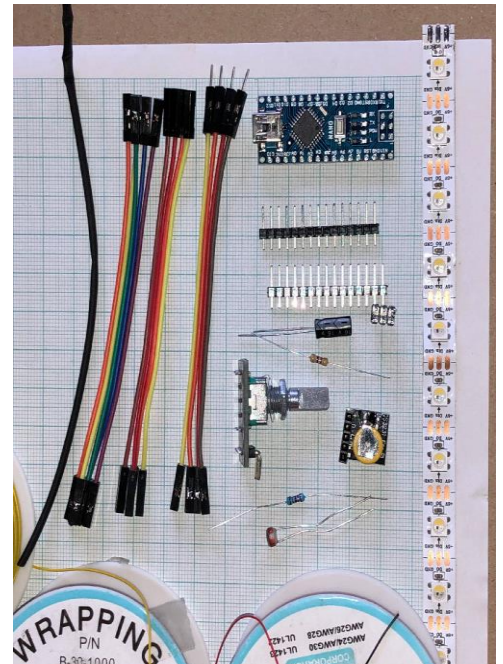
Benodigd is:

- WS2812 of SK6812 LED-strip met twaalf LEDs
- [Arduino Nano](#) of [Arduino Nano Every](#)
- DS3231 RTC-module.(1) DS3231 for Pi of DS3231 RTC
- 100 - 1000 uF condensator
- 330 of 470 ohm weerstand
- 330 of 470 ohm weerstand
- 22 Kohm weerstand
- LDR (lichtgevoelige weerstand)
- Rotary encoder
- 9 Dupont-kabeltjes vrouw-vrouw
- 4 Dupont-kabeltjes man-vrouw
- Isolatieband of krimpkous met diameter van ongeveer 2 mm
- Een paar stukjes draad
- Soldeerbout en soldeer

(1) In de kleinste klok past alleen de "DS3231 for Pi" klokmodule.

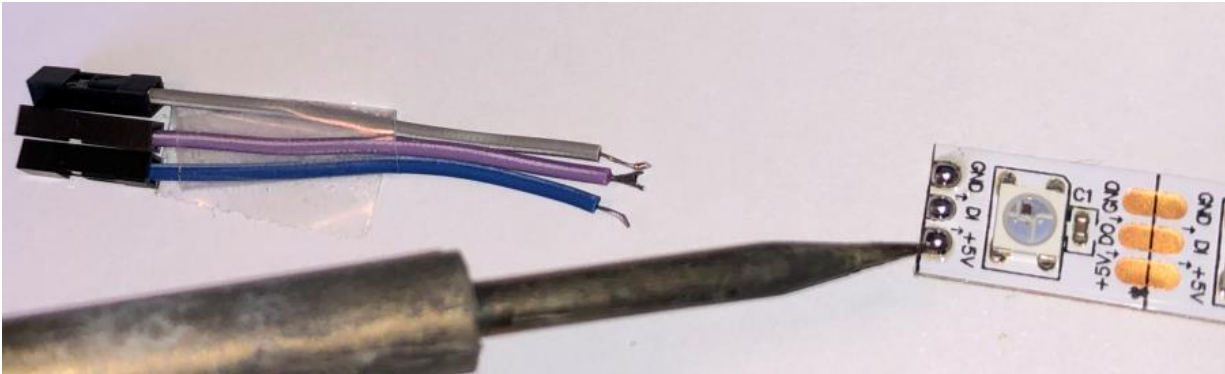
Hieronder is het aansluitschema getekend.

Bij [Fritzing](#) kan het programma gedownload worden om deze schema's te ontwerpen
Click op het plaatje voor het ontwerpbestand dat in het Fritzing-programma kan worden ingelezen en bewerkt.

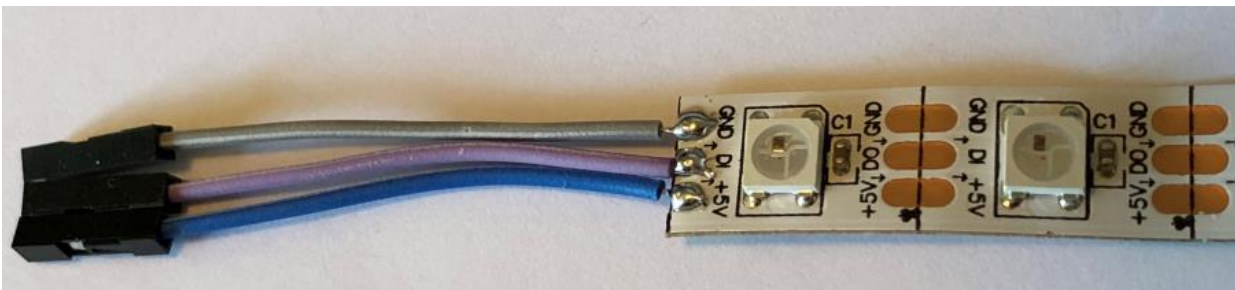


Fibonacci-vierkante kokerklok

- Breng soldeer aan op de drie aansluitingen aan de LED-zijde waar de GND-aansluiting boven ligt.



- Soldeer de drie Dupont-draden van ongeveer 5 cm lengte vast.



De weerstand is een extra beveiliging en kan eventueel weggelaten worden.

- Knip de middelste draad, verbonden met Di, door en soldeer hier de 470 ohm weerstand tussen.

Di is "Data in" en Do aan de andere kant van de LED is "Data out".

Dit is dus de signaaldraad die aan de LED's doorgeeft welke kleur zij moeten aannemen. In de LED zie je de kleine chip zitten die dat allemaal regelt.

De condensator kan eventueel weggelaten worden. De strip is relatief kort en trekt bij het aanzetten niet veel stroom.

Als de verlichting niet constant is en licht knippert dan kan de condensator dit verhelpen. Een zwaardere voeding waarschijnlijk ook.

- Soldeer aan de GND- en +5V-aansluiting de condensator. Langste poot van de condensator aan de 5V.

Aan de GND-kant moet op de condensator dan een grijze streep zichtbaar zijn met - - er op.

De condensator vangt de golf elektronen op die bij het aanzetten van de stroom de LED-strip binnenstroomt. De condensator dempt deze golf waardoor de LEDs bij het aanzetten minder belast worden.

- Breng een druppel lijm aan op de aansluiting zodat lostrekken voorkomen wordt.

Op de foto zijn nog extra draden gesoldeerd. Een GND (aardendraad) kan voor de LDR-aansluiting worden gebruikt.

Er is namelijk een GND-aansluiting tekort op de Arduino. In het schema is te zien dat twee draden naar de GND lopen.

Fibonacci-vierkante kokerklok

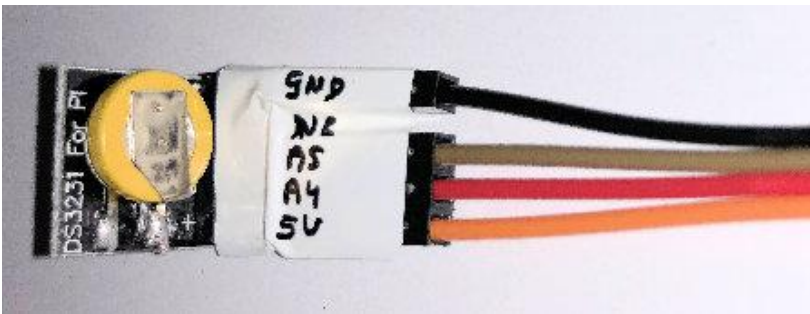
Het is verstandig om de draden te labelen. Dit voorkomt fouten bij het aansluiten en is praktisch mocht er ooit iets later losraken of vervangen moeten worden.

- Label alle draden dus zoals aangegeven in het Fritzing-ontwerpschema.

Het klopt dat de DS3231-klokmodule op 3.3V wordt aangesloten maar kan ook op 5V worden aangesloten.

De rotary ontvangt zijn voeding van een digitale pin die in de software "aan" wordt gezet.
`pinMode(EncoderPow, OUTPUT);`
`digitalWrite(EncoderPow,HIGH);` Er staat dan 5V op pin 2 en daar kan veilig 20 mA stroom van worden getrokken

De klok zou eventueel ook met een digitale pin kunnen worden gevoed.



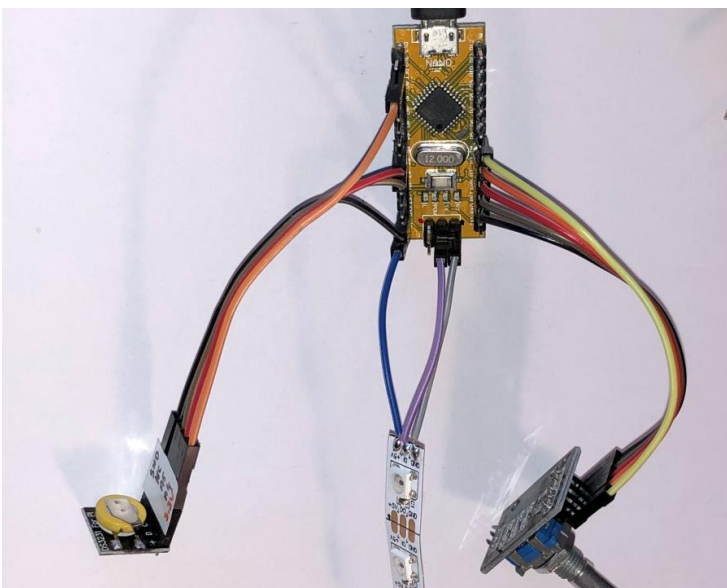
- Sluit de

LED-strip en eventueel de LDR aan zodat de elektronica getest kan worden.

Op de LED-strip tussen GND en 5V is de weerstand meer dan 1 Mohm (1 miljoen ohm).

Tussen 5V en GND pennen op de Arduino Nano is het meer dan 10 kOhm (10,000 Ohm).

- Sluit de Rotary en RTC-klokmodule aan volgens het Fritzing-schema.



- Sluit de USB-stekker aan op een 5V-voeding.

Wacht een paar seconden. Geen brandlucht en de LEDs gaan branden?

Fibonacci-vierkante kokerklok

De software

Vorbereiding IDE

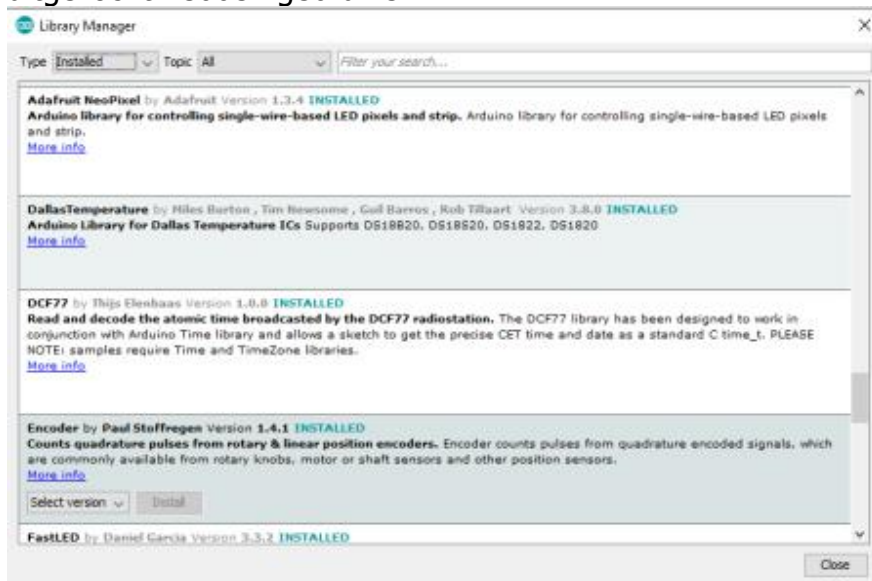
- Download de Arduino IDE van:
<https://www.arduino.cc/en/Main/Software>



Download the Arduino IDE



De software maakt gebruik van libraries. Dit is software door anderen geschreven welke functies bevat waar je anders maanden mee bezig was om uit te zoeken en te programmeren. Nu kan je dat werk dat anderen voor je uitgezocht hebben gebruiken.



De Arduino-programmeeromgeving biedt een groot scala aan libraries aan.

Om deze Fibonacci-kloksoftware te kunnen aanpassen moet in de Arduino IDE de volgende libraries geïnstalleerd worden:

dit gaat als volgt:

- In het IDE-menu open: Sketch->Include library -> manage libraries

- Zoek de volgende libraries en installeer ze

Adafruit Neopixel

Encoder by Paul Stofregen

Keypad by Mark Stanley, Alexander Brevig

RTCLib bij Adafruit

Wat ook kan is alle libraries van deze [ZIP-file](#) kopiëren naar de folder waar de sketches staan.

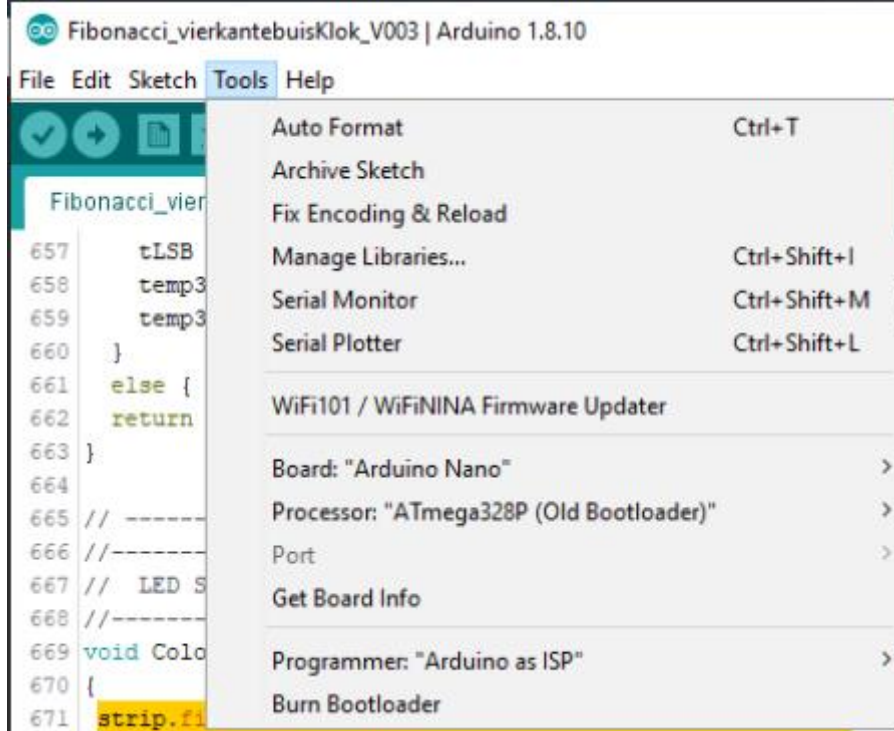
In deze sketchfolder bevindt zich een folder "libraries". Kopieer de inhoud van de ZIP-file hierin.

Er zijn veel soorten Arduino's. wij gebruiken de Arduino Nano

In het menu: Tools -> board kan je de Arduino Nano selecteren

Fibonacci-vierkante kokerklok

Kies bij Processor ATMEGA328p (Old bootloader)



Om de code in de Arduino Nano te plaatsen moet een Arduino via een de USB-kabel aan de PC worden aangesloten.

Port wordt dan zwart en er kan een compoort worden aangevinkt

Als daarna de pijl naar rechts onder edit in het menu wordt gedrukt wordt het programma in de Arduino nano geladen.

Het programma en zijn algoritmen

Met `#define` kunnen de gebruikte onderdelen geselecteerd worden worden.

Door twee slashes `//` voor een `#define` te zetten wordt de optie uitgeschakeld.

In het programma zijn de stukken die coderen voor een onderdeel te herkennen aan de programmaregels tussen `#ifdef` en `#endif`.

Het stukje code wordt niet gecompileerd in de software als het niet met een `#define` gedefinieerd is.

```
#ifndef ROTARYMOD
#include <Encoder.h> // For rotary encoder
#endif ROTARYMOD
```

De software ondersteunt WS2812 of SK6812 LED-strips. Kies een van deze.

Daarnaast kan er gekozen worden tussen een draaiknop (ROTARYMOD) of een keypad (KEYPAD).

De klok kan zonder DS3231-tijdmodule werken. Het werkt dan met de klok van de processor. Deze processorklok heeft een afwijking van seconden per dag. Dit is alleen handig om de software te testen.

Een DS3231-module heeft een afwijking van slechts een minuut per jaar.

Als `#define MOD_DS3231` is uitgezet wordt de processorklok gebruikt.

Fibonacci-vierkante kokerklok

```
//-----  
// ARDUINO Definition of installed modules  
//-----  
//#define LED2812           // Use  RGB LED strip WS2812  
#define LED6812           // Use RGBW LED strip SK6812  
#define ROTARYMOD         // Use rotary encoder  
//#define KEYPAD           // Use a 3x4 keypad  
#define MOD_DS3231       // DS3231 RTC module installed  
//-----  
// ARDUINO Includes defines and initialisations  
//-----  
#ifndef ROTARYMOD  
#include <Encoder.h>      // For rotary encoder  
#endif ROTARYMOD  
#include <Wire.h>         // Communication with  
#include <RTClib.h>      // For RTC module  
#include <EEPROM.h>      // To store data in EEPROM  
#include <TimeLib.h>     // For time management  
#include <Adafruit_NeoPixel.h> // for LED strip WS2812 or SK6812  
  
#ifndef KEYPAD  
#include <Keypad.h>      // For 3x4 keypad  
#endif KEYPAD
```

Bij "Pin Assigments" wordt een leesbare naam voor een pin gedefinieerd.

Digitale pinnen kunnen alleen aan of uit worden gezet of worden uitgelezen.

Bij analoge pinnen kunnen voltages tussen 0 en 5 V worden gemeten of aangestuurd worden met waarden tussen 0 en 1024 .

```
//-----  
// PIN Assigments  
//-----  
  
enum DigitalPinAssignments {  
  EncoderPow    = 2,           // give power to Encoder  
  clearButton   = 3,           // switch (labeled SW on decoder)  
  encoderPinA   = 4,           // right (labeled DT on decoder)  
  encoderPinB   = 5,           // left (labeled CLK on decoder)  
  EmptyD06      = 6,           // EmptyD06  
  EmptyD07      = 7,           // EmptyD07  
  EmptyD08      = 8,           // EmptyD08  
  EmptyD09      = 9,           // EmptyD09  
  EmptyD10      = 10,          // EmptyD10  
  LED_PIN       = 11,          // Pin to control colour SK6812 WS2812 LEDs  
  EmptyD12      = 12,          // EmptyD12  
  secondsPin    = 10};        // if set to 13 led will blink on board  
                               // Analogue hardware constants ----  
  
enum AnaloguePinAssignments {  
  PhotoCellPin  = 2,           // LDR pin  
  EmptyA3       = 3,           // EmptyA3  
  SDA_pin       = 4,           // SDA pin  
  SCL_pin       = 5};         // SCL pin
```

Na deze "Pin Assigments" volgen in de software nog meerdere definities van variabelen.

De Arduino gaat na de initialisaties de functie setup() uitvoeren.

Hierin worden allerlei functies van de libraries aangezet en de analoge en digitale pinnen gedefinieerd voor in- of uitvoer.

Fibonacci-vierkante kokerklok

Na de setup() wordt de functie loop() zoals de naam doet vermoeden eindeloos van begin tot eind in een lus doorlopen.

Dit is het hart van de coding waar vanuit alles wordt aangestuurd en berekend.

```
//-----  
// ARDUINO Loop  
//-----  
void loop(void)  
{  
  SerialCheck();  
  if(Demo) Demomode();  
  else  
  {  
    EverySecondCheck();  
    if(!KeyInputactivated) EveryMinuteUpdate(); // if keyboard  
    input then do not update display  
#ifndef ROTARYMOD  
    RotaryEncoderCheck();  
#endif ROTARYMOD  
  }  
}  
//-----  
// ARDUINO Setup initialise the hardware  
//-----  
void setup() // initialise the hardware // initialize the appropriate pins  
as outputs:  
{  
  Serial.begin(9600); // setup the  
  serial port to 9600 baud  
  
  pinMode(secondsPin, OUTPUT);  
#ifndef ROTARYMOD  
  pinMode(encoderPinA, INPUT_PULLUP);  
  pinMode(encoderPinB, INPUT_PULLUP);  
  pinMode(clearButton, INPUT_PULLUP);  
  pinMode(EncoderPow, OUTPUT);  
  digitalWrite(EncoderPow,HIGH); // Provide the  
  rotary encoder with power  
  Tekstprintln("Rotary encoder enabled");  
#endif ROTARYMOD  
  
  strip.begin(); // Start  
  communication to LED strip  
  strip.setBrightness(BRIGHTNESS); // Set  
  brightness of LEDs  
  ShowLeds();
```

et cetera

De software kijkt razendsnel in de loop of er wat serieel wordt ingetikt, of de demo stand is aangezet en of er aan de draaiknop wordt gezeten.

Verder zijn er nog eens per seconde en eens per minuut acties die de overige aandacht van de software eist.

Elke seconde wordt de lichtintensiteit van de LDR uitgelezen. Deze intensiteit wordt gewogen met de vorige uitlezing zodat de lichtsterkte van de LEDs gedempt aangepast wordt.

Van de uitlezing van de LDR, die tussen 0 en 1024 ligt, wordt de wortel genomen zodat de lichtsterkte parabolisch wordt aangepast.

De klok verandert elke minuut de tijdsweergave. Elke minuut de worden de getallen

Fibonacci-vierkante kokerklok

1,1,2,3,5 random een keer gekozen uit een array gekozen en opgeteld. Als de waarde overeenkomt met de gewenste waarde dat worden deze getallen naar de klok gestuurd. Een processor is zo enorm snel dat deze methode eleganter werkt dan alle mogelijkheden in een lijst te stoppen. Zeker als geheugenruimte beperkend is.

De kleurenpalettes zijn in een array colors[][] vastgelegd.

Bijvoorbeeld pallette nummer 1:

```
{ white, red , yellow, blue }, // #1 Mondriaan
```

Als een compartiment leeg moet blijven wordt de kleur wit aangezet.

Is er een kleur voor een uur dan wordt hij rood. Als de minuut gekleurd moet worden dan wordt de kleur geel. Als het compartiment zowel rood als geel is wordt de kleur blauw.

De variable bits[] houdt dit bij. Het kan de waarde 0, 1, 2 of drie hebben.

Bediening

De klok wordt met de rotary bediend. De rotary kan gedrukt en gedraaid worden.

Door te drukken op de knop wordt de knop geactiveerd. Na 60 seconden wordt deze weer gedeactiveerd.

Een keer drukken geeft de mogelijkheid de uren te veranderen. De display wordt dan even rood. Door aan de knop te draaien worden de uren vooruit of achteruit gezet.

Een tweede druk binnen 60 seconden doet display geel kleuren waarna de minuten aangepast kunnen worden.

Een derde druk geeft de mogelijkheid de maximale helderheid aan te passen.

Daarna kunnen de diverse kleurplattes geselecteerd worden.

De instellingen worden bewaard als de stroom van de klok wordt gehaald.

Om de klok te resetten moet 20 keer of meer gedrukt worden of lang ingedrukt houden.

```
switch (NoofRotaryPressed) // No of times the rotary is pressed
{
case 1: ChangeTime = true; ColorLeds("",0,NUM_LEDS-1,0XFF0000); ShowLeds();
delay(1000); break; // Change the hours RED
case 2: ChangeTime = true; ColorLeds("",0,NUM_LEDS-1,0XFFFF00); ShowLeds();
delay(1000); break; // Change the minutes YELLOW
case 3: ChangeLightIntensity = true; ColorLeds("",0,NUM_LEDS-1,0XFFFFFF);
ShowLeds(); delay(1000); break; // Change intensity
case 4: DisplayPalette = 1; break;
case 5: DisplayPalette = 2; break;
case 6: DisplayPalette = 3; break;
case 7: DisplayPalette = 4; break;
case 8: DisplayPalette = 5; break;
case 9: DisplayPalette = 6; break;
case 10: DisplayPalette = 7; break;
case 11: DisplayPalette = 8; break;
case 12: DisplayPalette = 9; break;
case 13: DisplayPalette = 0; break;
case 14 ... 19: break;
default: NoofRotaryPressed = 0; Reset(); break;
}
```

Keypad

De 3x4 toetsen keypad in de huidige software is te groot om op de klok te plakken maar is te gebruiken voor een kast.

Fibonacci-vierkante kokerklok

Seriële bediening

Nadat de klok met een seriële kabel aan de PC wordt gekoppeld kan met de Arduino-IDE software de klok bedient worden.

Door 'i' in te voeren komt er een menu in de "serial monitor" van de Arduino-IDE.

Elke dertig seconden wordt de tijd geprint en de gemeten lichtsterkte.

Met een aangesloten DS3231-tijdmodule wordt ook de gemeten temperatuur +/-5C getoond.

Sensor:683 Min:157 Max:736 Out: 41=16% Temp:22C

Sensor: is de actuele meting tussen 0 en 1024 bits.

Min: is de laagst gemeten bits en Max: de hoogste.

Out: is de berekende lichtintensiteit (0-255). In dit geval 16%.

Het menu na drukken van I is als volgt:

Enter Time as: hhmm (1321) or hhmmss (132145)

D for Demo mode

I for info

Lnn (L5) lowest intensity (1-255)

Mnn (M90) light intensity (1%-250%)

Pn (P1) to select a palette (0-9)

Max brightness: 20%

Min brightness: 10 bits

Number of LEDs: 12

De laagste lichtintensiteit dat naar de klok wordt gestuurd kan met L worden ingesteld.

De klok gaat dan niet zachter branden dan die waarde als het compleet donker is.

L5 is een mooie waarde.

Met M kan de maximale sterkte worden ingesteld. Soms is de klok te fel of te zwak voor de plek waar hij staat.

M80 is de default waarde.

Met P kan de kleurpalette worden gekozen. P1 is standaard.

Met D gaat de klok in demo mode tot D weer ingevoerd wordt of de klok uit wordt gezet.

Succes

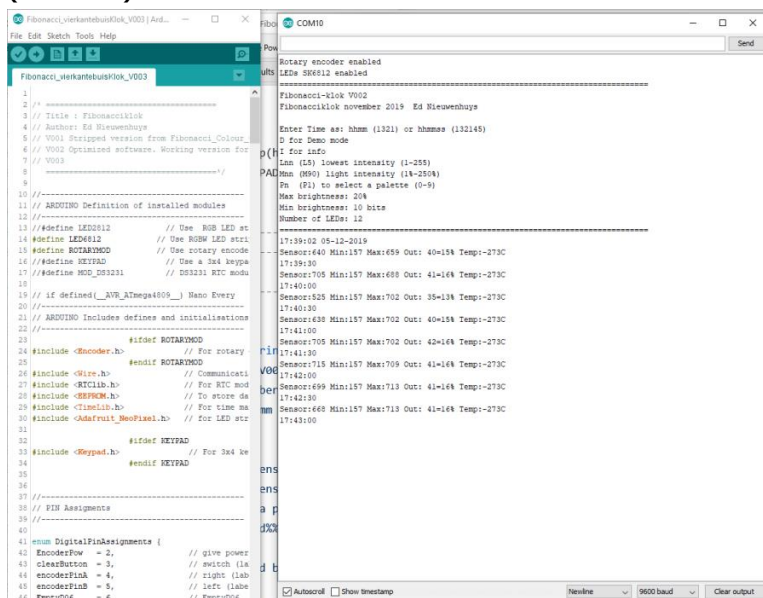
Ed Nieuwenhuys

[Software op Github](#) (zoek in Google "ednieuw gitub")

Mail:ednieuw@xs4all.nl

Of zoek in de internetpagina in Google met: ednieuw stokklok

31 jan 2020



The image shows two windows from the Arduino IDE. The left window displays the source code for 'Fibonacci_vierkantebusklok_V003'. The code includes comments for various settings like LED intensity (L5), light intensity (M90), palette selection (P1), and demo mode (D). It also includes hardware definitions for the Arduino board and the DS3231 module. The right window is the 'Serial Monitor' showing the output of the program, which includes the current time, sensor readings, and the calculated light intensity (Out: 41=16%).