Dries Kennes & Michiel Bellekens | Project II | 2015-2015

Smartclock

Project II – ICT-Elektronica



# Voorwoord

Wij zijn Dries Kennes en Michiel Bellekens, studenten ICT-Elektronica Fase 2 aan Thomas More Mechelen op Campus De Nayer.

Dit project heeft als doel het maken van een smartclock. Zoals de naam reeds doet vermoeden is het een klok/wekker met enkele slimme extra functies die nuttig kunnen zijn voor de gebruiker. Zo is de belangrijkste feature dat de gebruiker kan worden gewekt aan de hand van zijn/haar kalender.

De gebruiker kan via een web interface de klok configureren naar zijn/haar eigen wensen. Ten eerste kan de gebruiker instellen hoe lang voor de eerste afspraak de wekker moet afgaan. Ten tweede kan men kiezen of men gewekt wenst te worden via een muziekfile, muziekstream of via voorgeprogrammeerde geluiden. Ten derde kan de gebruiker ook een minimum en maximum wektijd instellen. Dit betekent dat onafhankelijk van de afspraken de wekker nooit vroeger dan het minimum en later dan het maximum mag afgaan. Tot slot zijn er nog enkele instellingen voor de lay-out op het scherm zoals het formaat en de font size van de tijd en de datum.

Ons contacteren kan via email, [dries.kennes@student.thomasmore.be](mailto:dries.kennes@student.thomasmore.be) , michiel.bellekens@student.thomasmore.be, of via de website, [dries007.net](http://dries007.net).

Inhoudsopgave

[1 Voorwoord 2](#_Toc450915498)

[2 Hardware 5](#_Toc450915499)

[2.1 Raspberry Pi Zero Essentials kit 5](#_Toc450915500)

[2.2 I2C RTC (3.3 V DS3231) 5](#_Toc450915501)

[2.3 2.4” 240x320 spi tft lcd (3.3 V) 6](#_Toc450915502)

[2.4 Spanningsregelaar (MCP1703) 7](#_Toc450915503)

[2.5 Levelshifter (AN10441) 7](#_Toc450915504)

[2.6 Audio stereo DAC (PCM 5102A 3.3 V) 8](#_Toc450915505)

[2.6.1 Intermezzo I2S 8](#_Toc450915506)

[2.7 Audio amplifier (TPA2016D2 5V) 9](#_Toc450915507)

[2.8 Overige onderdelen 9](#_Toc450915508)

[2.8.1 Rotary encoder 9](#_Toc450915509)

[2.8.2 Buzzer 9](#_Toc450915510)

[2.8.3 NOR gate flipflop 10](#_Toc450915511)

[2.8.4 Supercap 10](#_Toc450915512)

[2.8.5 WS2812 led’s 10](#_Toc450915513)

[2.9 De PCBS 12](#_Toc450915514)

[3 Software 14](#_Toc450915515)

[3.1 Main code (app.py) 15](#_Toc450915516)

[3.1.1 Google calendar authenticatie 15](#_Toc450915517)

[3.1.2 Netwerk en acces point 15](#_Toc450915518)

[3.1.3 LCD aansturen 15](#_Toc450915519)

[3.1.4 Web interface verwerking/flask 15](#_Toc450915520)

[3.1.5 RTC aansturen 15](#_Toc450915521)

[3.1.6 Date/time 15](#_Toc450915522)

[3.1.7 Rotary encoder 15](#_Toc450915523)

[3.1.8 WS2812 led’s aansturen 15](#_Toc450915524)

[3.1.9 Web interface 16](#_Toc450915525)

[3.1.9.1 Status tab 16](#_Toc450915526)

[3.1.9.2 Wifi settings tab 16](#_Toc450915527)

[3.1.9.3 Clock settings tab 16](#_Toc450915528)

[3.1.9.4 Google calendar tab 16](#_Toc450915529)

[4 Budget/kostenraming 17](#_Toc450915530)

[5 Besluit 18](#_Toc450915531)

[6 Bijlagen 19](#_Toc450915532)

[6.1 PCB Schema’s & Lay-out 20](#_Toc450915533)

[6.2 Scripts 21](#_Toc450915534)

[6.3 Broncode (app.py) 22](#_Toc450915535)

[6.3.1 De website 23](#_Toc450915536)

[6.3.1.1 Index.php 23](#_Toc450915537)

[6.3.1.2 Hardware.php 23](#_Toc450915538)

[6.3.1.3 Software.php 23](#_Toc450915539)

[6.3.1.4 Pictures.php 23](#_Toc450915540)

[6.3.1.5 About.php 23](#_Toc450915541)

[6.4 Foto’s 23](#_Toc450915542)

# Hardware

## Raspberry Pi Zero Essentials kit

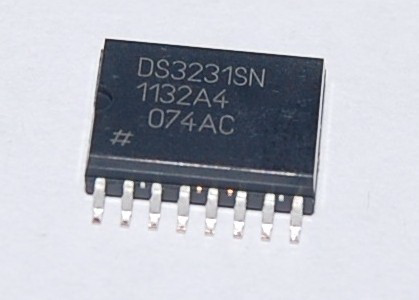
Het embedded platform dat wordt gebruikt in dit project is de Raspberry Pi zero. Dit is gekozen om een aantal verschillende redenen. Ten eerste biedt de zero een zeer goede ondersteuning voor verschillende communicatiemogelijkheden zoals SPI, I2S en I2C maar ook voor het opzetten van draadloze netwerken of webservers. Ten tweede is de kostprijs ook zeer aantrekkelijk aangezien de Pi Zero Essentials kit slechts €8 kost en alle basis benodigdheden bevat.



Figuur 1: De inhoud van de Raspberry Pi Zero Essentials kit.

## I2C RTC (3.3 V DS3231)

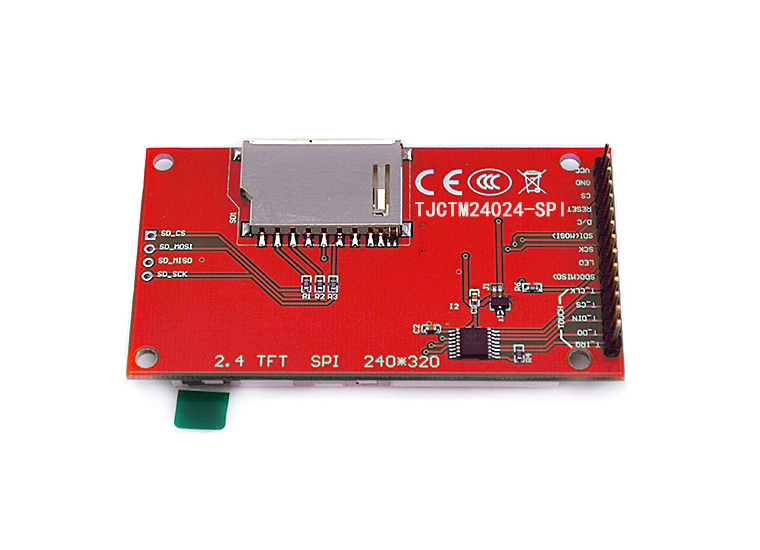
De eerste belangrijke component voor de werking van de Smartclock is de DS3231 real time clock. Deze chip werd gekozen voor zijn hoge nauwkeurigheid en zijn instelbaarheid via de ingebouwde I2C interface die ook wordt ondersteund in de Linux kernel. Zijn hoge nauwkeurigheid dankt de chip aan de ingebouwde temperatuur gecompenseerde kristal oscillator. Een extra reden voor de keuze van deze chip is de mogelijkheid om een batterij toe te voegen als back-up voeding. Het overschakelen naar de batterijspanning gebeurt automatisch wanneer de chip detecteert dat zijn voedingspanning wegvalt. Dit is handig om te kunnen garanderen dat het alarm toch nog minstens 1 keer kan afgaan nadat de hoofdvoeding is uitgevallen. Naast de tijd in uren, minuten en seconden (24 uurs of 12 uurs met AM/PM) wordt ook de datum bijgehouden. Deze datum wordt automatisch aangepast, rekening houdend met het aantal dagen in elke maand en ook correctie voor schrikkeljaren. De klok geeft toegang tot 2 time-of-day alarmen die bij een alarmconditie de INT (interrupt) pin gaan aansturen.



Figuur 2: De package van de DS3231 chip

## 2.4” 240x320 spi tft lcd (3.3 V)

Om de klok en eventueel andere informatie te kunnen tonen aan de gebruiker hebben we een 2.4” 240x320 SPI tft display gebruikt. Datasheets voor deze display zijn moeilijk te vinden. Hierdoor moesten we ons baseren op de weinige informatie die we wel konden vinden en voor de rest wachten tot de displays toekwamen.



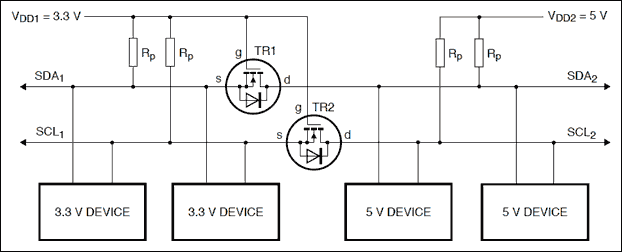
Figuur 3: De SPI 2.4" 240x320 tft lcd display voor -en achteraanzicht.

## Spanningsregelaar (MCP1703)

Aangezien onze voeding 5V bedraagt en een deel van onze componenten zoals de lcd op 3.3 V werken moeten we de spanning tot naar 3.3 V kunnen regelen. De MCP1703 is een CMOS low drop-out (max 650mV) spanningsregelaar die 250 mA kan leveren en zelf slechts 2 µA verbruiken . Zijn inputspanning ligt tussen 2.7 V – 16 V en zijn output spanning is dus 3.3 V.

## Levelshifter (AN10441)

De Raspberry pi Zero gebruikt 0 V en 3.3 V als GPIO level. Om te kunnen communiceren met de andere componenten op 5 V zoals de RTC hebben we levelshifters nodig. Aangezien de communicatie over bv. I2C bi-directioneel is, moeten de levelshifters ook bi-directioneel zijn en snel kunnen werken. De makkelijkste manier is om MOSFET’s op elke lijn te plaatsen. De MOSFET’s werken bi-directioneel door zijn 3 mogelijke states. State 1 is wanneer het 3.3 V gedeelte hoog wordt/is. In dit geval is de MOSFET niet in geleiding aangezien de drempelspanning tussen de gate en de source niet is bereikt. Doordat de MOSFET niet in geleiding is wordt de 5 V kant op zijn beurt ook hoog getrokken door zijn eigen pull-up weerstand. Beid kanten zijn dus hoog maar op een ander spanningsniveau. De tweede state is wanneer de 3.3 V kant wordt laag getrokken. In dit geval wordt de drempelspanning tussen de gate en de source wel overschreden waardoor de MOSFET in geleiding gaat. Hierdoor wordt het 5 V gedeelte ook laag getrokken. De derde state is wanneer de 5 V kant laag wordt getrokken. In dit geval zal de diode, ingebouwd in de MOSFET ervoor zorgen dat de 3.3 V kant laag wordt getrokken tot een level waarbij de drempelspanning wordt overschreden. Wanneer dit gebeurt, zal de MOSFET in geleiding gaan waardoor het 3.3 V gedeelte nog verder wordt laag getrokken.



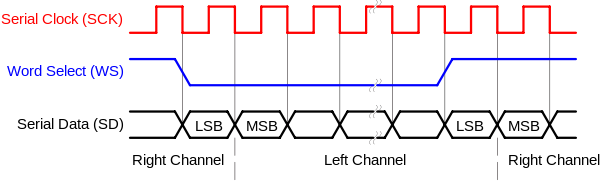
Figuur 4: Shiften tussen 3.3 V en 5 V op een I2C bus.

## Audio stereo DAC (PCM 5102A 3.3 V)

Om ook muziek te kunnen spelen wanneer de wekker afgaat, hebben we een Digital to analog converter nodig. Deze chip ondersteunt de I2S serial bus interface standaard die dient voor digitale audio (zie intermezzo I2S). Dit is handig aangezien de Raspberry Pi Zero deze standaard ondersteunt. De PCM 5102A chip heeft een stereo output, wat wil zeggen dat er een L (links) en R (rechts) kanaal is voor de audio.

### Intermezzo I2S

I2S staat voor Inter-IC Sound en is een seriële bus interface standard die wordt gebruikt om verschillende digitale audio devices te verbinden. De bus heeft minimum 3 lijnen: bit clock lijn, word clock lijn (WS of LRCLK) en een data lijn. De bit clock wordt gepulst voor elke bit op de datalijnen. De word clock laat het device weten voor welk kanaal (1 of 2 ) de huidige data is bedoelt. Wanneer de word clock laag is, is de data bedoelt voor het linker kanaal, anders voor het rechter kanaal.



Figuur 5: De timing van een I2S seriële bus interface.

## Audio amplifier (TPA2016D2 5V)

De TPA2016D2 is een stereo audio versterker die tot 2.8 W/kanaal kan leveren afhankelijk van de weerstand van de speakers. De chip bevat ook een Dynamic Range Compression (DRC) en Automatic Gain Control (AGC) functie. De DRC functie gaat dynamisch de range van het geluid beperken. Dit wil zeggen dat de harde geluiden boven een bepaalde waarde worden afgezwakt terwijl de waardes onder deze drempel ongewijzigd blijven. De belangrijkste functie van de DRC is het opvangen van te grote niveauverschillen in het geluid. De AGC gaat op zijn beurt ervoor zorgen dat de versterking automatisch wordt aangepast aan het ingangssignaal. Zwakkere signalen zullen dus harder worden versterk als de sterkere signalen. De versterker kan tussen -28dB en 30dB versterken op beide kanalen van het stereo signaal. Het instellen van de versterker kan door via I2C, 7 registers in te stellen. In deze registers kan bv. De versterking en de versterkingssnelheid worden ingesteld.

## Overige onderdelen

### Rotary encoder

Als input voor bv. het volume van het geluid of de helderheid van het scherm wordt gebruik gemaakt van een rotary encoder. De gebruikte rotary encoder heeft naast een rotatie functie ook een drukknop functie.



### Buzzer

De buzzer dient hoofdzakelijk als back-up wekker. Indien de voedingspanning wegvalt, zal door de back-up voeding van de RTC deze toch nog blijven werken. Tijdens normale werking wordt het alarm in de RTC afgezet vooraleer het zijn interrupt kan geven en zullen de speakers voor het alarm zorgen. Indien de Raspberry pi niet gevoed wordt, zal dit niet gebeuren waardoor de interrupt wel wordt gegeven en er een set plaats vindt van de flipflop. Dit zorgt er op zijn beurt voor dat de buzzer zal afgaan tot de gebruiker de flipflop reset via een aparte knop. De Raspberry pi kan de buzzer ook setten en resetten indien nodig.



### NOR gate flipflop

De flipflop wordt gebruikt om te kunnen bepalen hoelang de buzzer moet aflopen. Wanneer de flipflop wordt geset

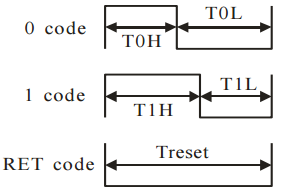
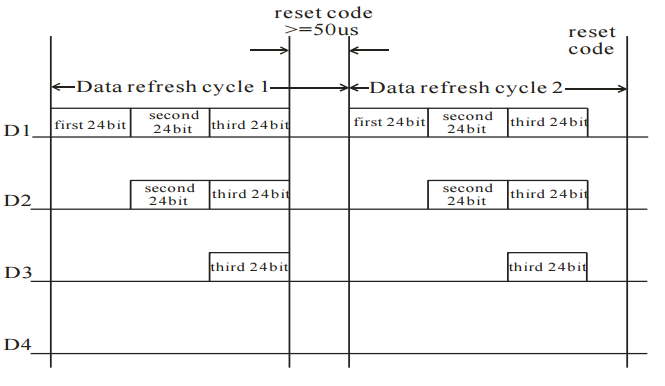
### Supercap

Als back-up voeding voor de basisfunctionaliteit van de klok wordt een supercap van 1.5 farad gebruikt. De RTC schakelt over naar de voeding van de supercap indien de voedingsspanning wegvalt.



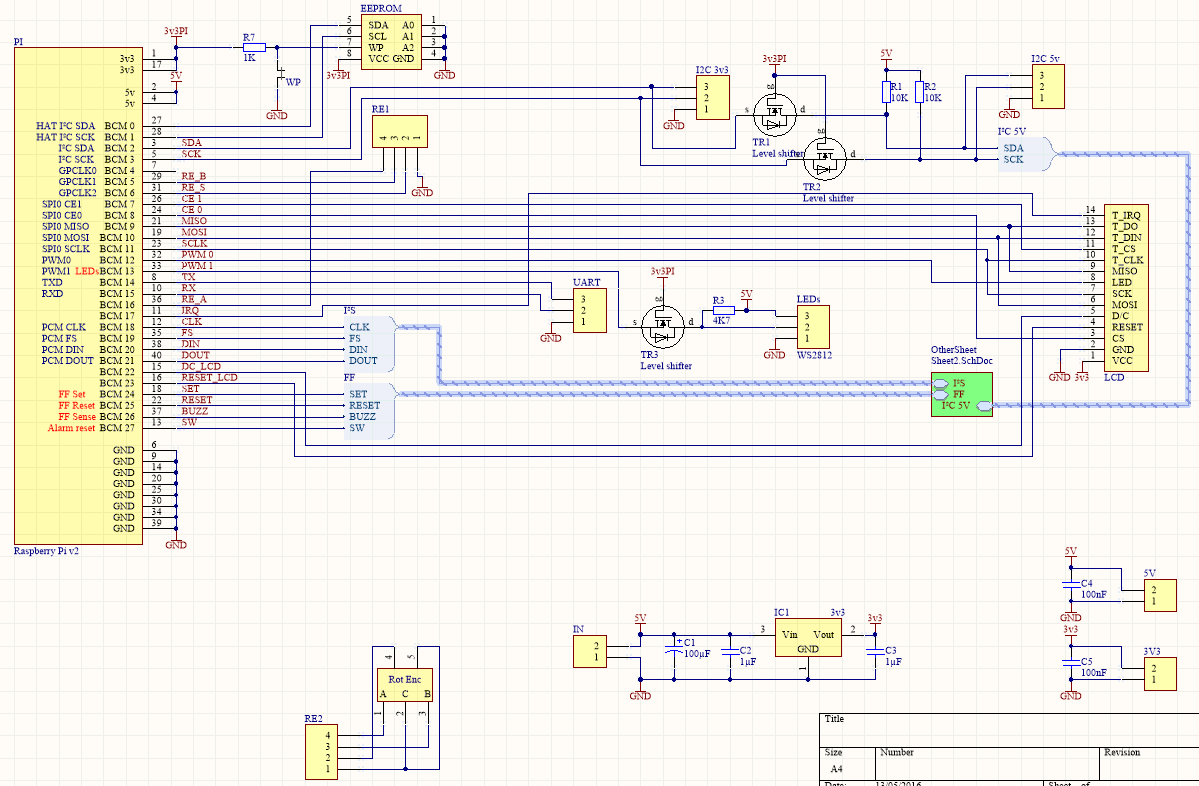
### WS2812 led’s

De WS2812 LEDs werken als een lange serie schakeling, waarbij elke led de eerste 24 bits gebruikt om zijn kleur in te stellen. De andere bits worden doorgegeven, zie onderstaande figuren. De volgorde van de bits is niet RGB, maar GRB, met hoogste bit eerst.

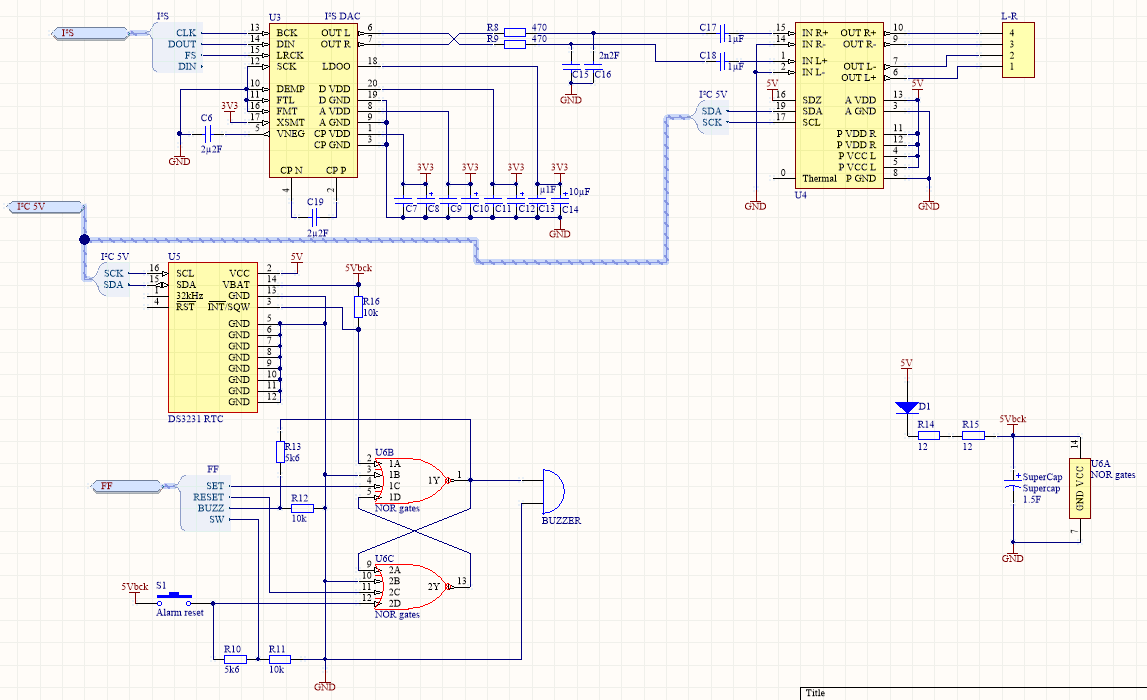


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code | Betekenis | Tijd | Min | Typ | Max |
| T0H | 0 code - high voltage time | 0,35 µs ±150 ns | 200 ns | 350 ns | 500 ns |
| T1H | 1 code - high voltage time | 0,70 µs ±150 ns | 550 ns | 700 ns | 850 ns |
| T0L | 0 code - low voltage time | 0,80 µs ±150 ns | 650 ns | 800 ns | 950 ns |
| T1L | 1 code - low voltage time | 0,60 µs ±150 ns | 450 ns | 600 ns | 750 ns |
| Reset | low voltage time | > 50µs | 50 µs |  |  |
| TH+TL | Cyclus time | 1,25µs ±600ns | 660 ns | 1250 ns | 1850 ns |

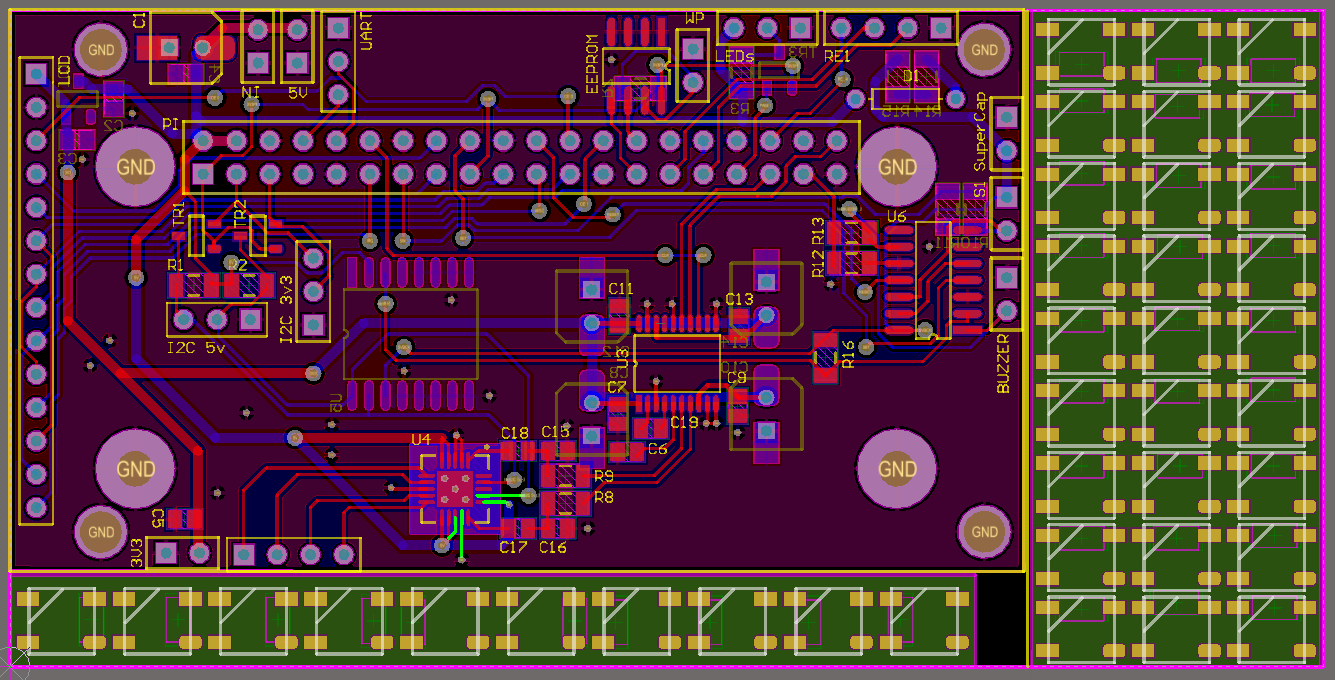
## De PCBS



Figuur 6: PCB schema sheet 1.



Figuur 7: PCB schema sheet 2.



Figuur 8: Het roze gedeelte is de PCB lay-out voor dit project. Om de resterende ruimte van de print niet verloren te laten gaan werden hier pads voor ws2812 led’s voorzien.

# Software

## Main code (app.py)

### Google calendar authenticatie

De Smartclock gebruikt de google calendar API om toegang te kunnen krijgen tot de kalender data van de gebruiker.

De eerste stap is het registreren van de Smartclock via de Google developers console. Tijdens het registeren moet de developer aanduiden tot welke onderdelen van google de toepassing wilt gebruiken. Na de registratie geeft google de developer een client ID en client secret die zeer belangrijk zijn verder in het authenticatie proces. De client ID en client secret zijn hetzelfde voor alle Smartclocks.

De tweede stap is het versturen van een http post met de client ID en met een lijst van scopes. De scope geeft aan welk level van toegang de toepassing wilt. De response van deze request is een json object dat 5 zaken bevat. De user code en verificatie URL moeten aan de gebruiker worden getoond. De device code, expire time en interval zijn later nodig bij het verkrijgen van een access token.

Ten derde moet de gebruiker de toepassing de toegang verlenen tot zijn kalender. Dit kan gebruiker doen door naar de verificatie URL te gaan en zijn getoonde user code in te geven. Wanneer dit is gebeurd zal google een pagina tonen waarin staat omschreven tot wat de toepassing toegang zal krijgen. De gebruiker kan deze toegang toestaan of weigeren.

Na het tonen van de user code en de verification URL kan de toepassing een access token proberen te verkrijgen. Dit gebeurt door een post request te doen met de device code, client ID en client secret en met tussen elke request zeker de tijd weergegeven door het interval uit stap 2. Zolang de gebruiker nog geen toegang heeft gegeven zal deze request een json object met een error teruggeven. Deze error kan verschillende zaken bevatten zoals slow down wanneer je een te klein interval gebruikt tussen de requests. Nadat de gebruiker de toegang heeft verleend zal het antwoord een json object zijn dat een access token, refresh token, token type en expiration time bevat. De access token en refresh token moeten worden opgeslagen voor langere tijd. De access token wordt gebruikt tijdens het opvragen van informatie uit de kalender. De refresh token is nodig bij het verkrijgen van een nieuwe access token na het verlopen van de vorige.

Het opvragen van kalender gebeurtenissen kan nu via een request met de access token naar de google calendar endpoint. Deze request zal een json object met alle data teruggeven die dan kan worden verwerkt in de toepassing.

Indien de access token is vervallen kan met de refresh token een nieuwe worden opgevraagd. Bij deze request moet ook de client ID en client secret worden meegestuurd.

De volledige handleiding over de google API is te vinden op : <https://developers.google.com/identity/protocols/OAuth2ForDevices>

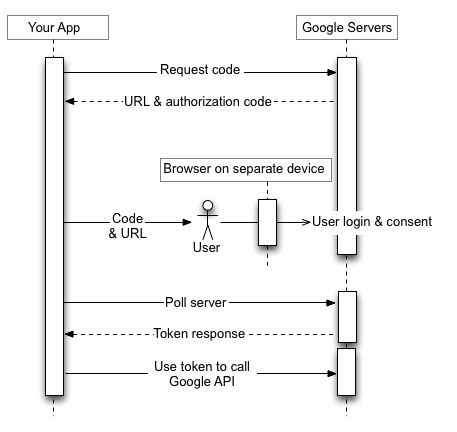


Figure 9:Schematische voorstelling van het stappenplan om toegang te krijgen tot de google calendar van de gebruiker.

### Netwerk en acces point

### LCD aansturen

### Web interface verwerking/flask

### RTC aansturen

### Date/time

### Rotary encoder

### WS2812 led’s aansturen

### Web interface

Voor de web interface is de HTML, CSS en javascript framework bootstrap gebruikt. Dit stelt ons in staat om op een snelle en relatief simpele manier een mooie website te maken die schaalbaar is voor verschillende toestellen. De web interface bestaat uit 1 webpagina met 4 verschillende tabs. Het nut van elke tab wordt hieronder kort uitgelegd.

#### Status tab

De status tab geeft de gebruiker wat informatie over de status van het toestel.

#### Wifi settings tab

Deze tab geeft een lijst van alle beschikbare netwerken. De gebruiker kan hier zijn netwerk kiezen en instellen.

#### Clock settings tab

De settings tab is de belangrijkste tab na het in gebruik nemen van het toestel. Hier kan de gebruiker de Smartclock instellen naar zijn eigen wensen. Ten eerste kan het format en de size van de tijd en de datum worden ingesteld. Ten tweede kan de gebruiker instellen hoelang voor de eerste afspraak de wekker moet afgaan. Ten derde kan ook optioneel een minimum en maximum wek tijd worden ingesteld. Deze waarden willen zeggen dat de wekker nooit vroeger dan het minimum en later dan het maximum mag afgaan. Een vierde instelling bepaalt of de wekker enkel in de week, weekend of elke dag mag afgaan. Ten vijfde kan de gebruiker kiezen of de dag wordt weergegeven of niet en indien het wordt getoond in welke size dit moet. De laatste instelling bepaalt welk type van alarm er moet worden gebruikt tijdens het wekken. De gebruiker heeft hierbij de keuze uit een muziekfile, muziekstream of 1 van de ingebouwde geluiden.

#### Google calendar tab

In de google calendar tab wordt de verification URL en de user code getoond. Deze heeft de gebruiker nodig om de Smartclock toegang te geven tot zijn google calendar?

# Budget/kostenraming

Dit is een kostenraming van de componenten voor dit project (voor 1 product):

|  |  |
| --- | --- |
| Naam | Kostprijs |
| Wifi dongle | € 2 |
| Buzzer | € 3 |
| USB adapter | € 1 |
| RTC | € 0.50 |
| Supercap | € 2 |
| Levelshifters | € 0.20 |
| 2 x speaker | € 2.50 |
| 3.3V spanningsregulator | € 0.50 |
| Micro SD | € 8 |
| Led’s | € 0.50 |
| Power adapter | € 2 |
| Amplifier | € 2.50 |
| Rotary encoders | € 0.50 |
| PCB | € 3.50 |
| Raspberry Pi Zero essentials kit | € 8 |
| LCD scherm | € 7 |
| DAC | € 2.50 |
| 8 x WS2812 led’s | € 0.80 |
| **Totaal** | **€ 47** |

# Besluit

# Bijlagen

## PCB Schema’s & Lay-out

## Scripts

## Broncode (app.py)

De volledige broncode is ook beschikbaar op [github.com/\*](https://github.com/dries007/ModernMasterMind).

### De website

Ook voor de website werd het bootstrap framework gebruikt.

#### Index.php

#### Hardware.php

#### Software.php

#### Pictures.php

#### About.php

## Foto’s