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).

# Inhoud

[1 Voorwoord 1](#_Toc450549567)

[2 Inhoud 2](#_Toc450549568)

[3 Hardware 3](#_Toc450549569)

[3.1 Raspberry Pi Zero essential kit 3](#_Toc450549570)

[3.2 I2C RTC (DS3231) 3](#_Toc450549571)

[3.3 2.4” 240x320 spi tft lcd 4](#_Toc450549572)

[3.4 De PCBs 4](#_Toc450549573)

[3.5 Overige onderdelen 5](#_Toc450549574)

[4 Software 6](#_Toc450549575)

[4.1 De memory map 6](#_Toc450549576)

[4.2 Het WS2812 protocol 7](#_Toc450549577)

[4.3 De AVR software 8](#_Toc450549578)

[4.3.1 Het hoofdprogramma (main) 8](#_Toc450549579)

[4.3.2 De interrupt routine 8](#_Toc450549580)

[4.3.3 De LCD driver 8](#_Toc450549581)

[4.3.4 Debug code 8](#_Toc450549582)

[4.3.5 De WS2812 driver 9](#_Toc450549583)

[4.4 De SC12 Software 10](#_Toc450549584)

[4.4.1 Het hoofdprogramma 10](#_Toc450549585)

[4.4.2 IP2LCD 10](#_Toc450549586)

[4.4.3 De webpagina’s 11](#_Toc450549587)

[5 Budget 12](#_Toc450549588)

[6 Besluit 13](#_Toc450549589)

[7 Bijlagen 14](#_Toc450549590)

[7.1 PCB Schema’s & Layout 14](#_Toc450549591)

[7.2 Scripts 15](#_Toc450549592)

[7.2.1 make.bat 15](#_Toc450549593)

[7.2.2 compile.sh 16](#_Toc450549594)

[7.3 Broncode 17](#_Toc450549595)

[7.3.1 mastermind.h 17](#_Toc450549596)

[7.3.2 mastermind.c 18](#_Toc450549597)

[7.3.3 ramdump.h 19](#_Toc450549598)

[7.3.4 ramdump.c 20](#_Toc450549599)

[7.3.5 HTML template 21](#_Toc450549600)

[7.3.6 AVR.h 21](#_Toc450549601)

[7.3.7 AVR.c 21](#_Toc450549602)

[7.4 Foto’s 21](#_Toc450549603)

# Hardware

## Raspberry Pi Zero essential 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.



Figure 1: De inhoud van de Raspberry Pi Zero essantials 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.

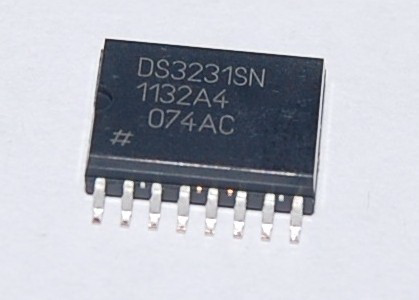


Figure 2: De package van de DS3231 chip

## 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.

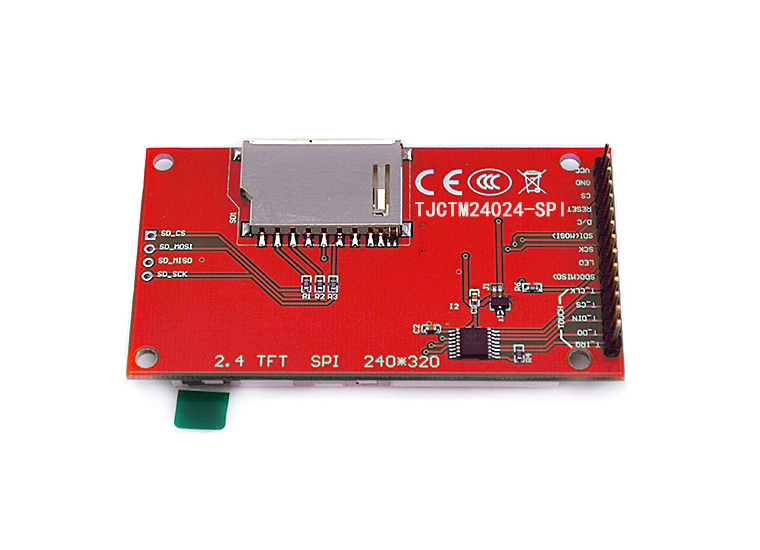


Figure 3: De SPI 2.4" 240x320 tft lcd display voor -en achteraanzicht.

## 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 dropout (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 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.

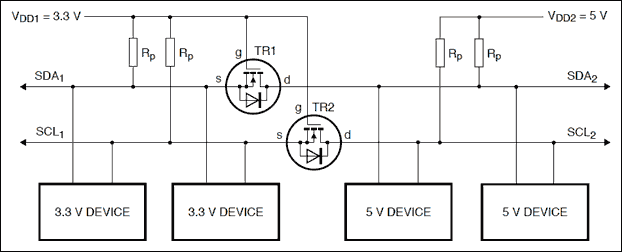


Figure 4: Shiften tussen 3.3 V en 5 V op een I2C bus.

## 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.

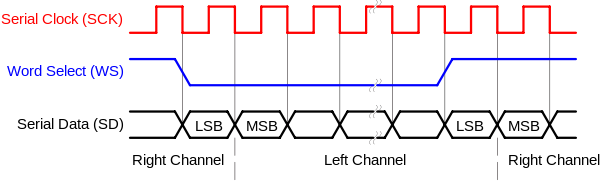


Figure 5: De timing van een I2S seriële bus interface.

## De PCBs

## Overige onderdelen

# Software

## De memory map

## Het WS2812 protocol

## De AVR software

### Het hoofdprogramma (main)

### De interrupt routine

#### 0x00 of een waarde die niet voorkomt in de tabel– Geen functie

#### 0x01 – Update LEDs

#### 0x02 – Print LCD karakter buffer

#### 0x03 – Stuur LCD instructie

#### 0x04 – Stuur clear instructie en print LCD karakter buffer

#### 0x05 – Zet LCD cursor positie op LCD commando

#### 0x06 of 0x07 –LCD backlight aan of uit

### De LCD driver

### Debug code

### De WS2812 driver

## De SC12 Software

### Het hoofdprogramma

### IP2LCD

### De webpagina’s

#### /home

#### /pickUsername

#### /reset

#### /start

#### /play

##### Gok subroutine

# Budget

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 |
| Level shifters | € 0.20 |
| 2 x speaker | € 2.50 |
| 3.3V spanningsregulator | € 0.50 |
| Micro SD | € 8 |
| Led’s | € 0.50 |
| Power adapter | € 2 |
| Amplifiers | € 2.50 |
| Rotary encoders | € 0.50 |
| PCB | € 3.50 |
| Raspberry Pi Zero essentials kit | € 8 |
| LCD scherm | € 7 |
| DAC | € 2.50 |
| **Totaal** | **€ 46.2** |

# Besluit

# Bijlagen

## PCB Schema’s & Layout

## Scripts

### make.bat

### compile.sh

## Broncode

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

### mastermind.h

### mastermind.c

### ramdump.h

### ramdump.c

### HTML template

### AVR.h

### AVR.c

## Foto’s