

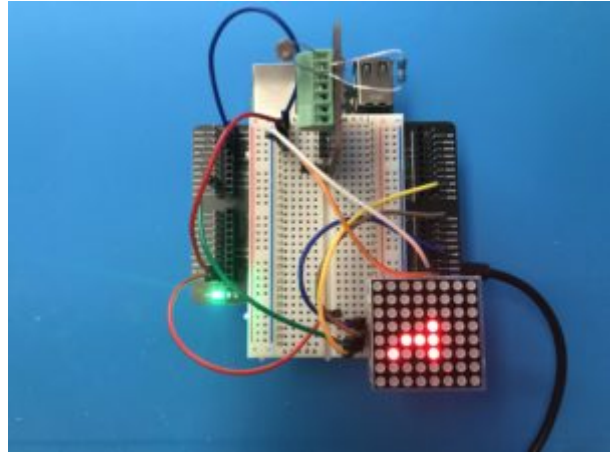
# MAX7219 analoge klok op 8x8 dot-matrix display

 [raspberrypi.a-vision.solutions/max7219-8x8-dot-matrix-licht-gevoelige-analoge-klok](https://raspberrypi.a-vision.solutions/max7219-8x8-dot-matrix-licht-gevoelige-analoge-klok)

8 februari 2021

Voor dit project gaan we een MAX7219 8×8 dot-matrix display aansturen (analoge klok). De intensiteit gaan we aansturen aan de hand van een lichtmeting.

We maken het aansturingsprogramma met Python 3.



## Installatie

### Je hebt nodig-

- Raspberry Pi 3/4/400/Zero (wordt niet meegeleverd bij de experimenteer set)
- [MAX7219 8×8 dot-matrix display](#).
- [ADS1015 4-kanaals analoog naar digitaal omzetter](#)
- [Raspberry Pi 400-pin experimenteer bord](#)
- 100Kohm weerstand
- Lichtgevoelige weerstand ([LDR](#))

### Voordat je begint even dit-

De hardware aansturing van de MAX7219 maakt gebruik van de SPI interface, en de aansturing van de ADS1015 gebruikt de I<sup>2</sup>C interface.

Om deze te kunnen gebruiken moeten ze worden aangezet op de Raspberry Pi. Dit kan worden gedaan met het [raspi-config programma op de Raspberry Pi](#).

We zullen ook SSH aanzetten op de Raspberry Pi om via Windows de Raspberry Pi te kunnen benaderen/programmeren.

Verder wordt in deze uitleg gebruik gemaakt van het [Windows programma Thonny](#) voor het programmeren met Python.

### MAX7219 8×8 dot-matrix display

Voor het aansturen van de MAX7219 8×8 dot-matrix display maken we gebruik van de [LUMA Display drivers voor MAX7219](#).

Het is belangrijk deze te installeren en te testen alvorens dit project verder te

programmeren.

Voor de installatie wordt verwezen naar. – <https://luma-led-matrix.readthedocs.io/en/latest/install.html>

Na installatie wordt aangeraden om met behulp van het test programma `examples/matrix_demo.py` te controleren of het goed functioneert.

---

## ADS1015 analoog naar digitaal omzetter

---

Voor het uitlezen van de ADS1015 wordt gebruik gemaakt van de Python Package `adafruit-ads1x15`. Deze is eenvoudig te installeren door de instructies te volgen op. – <https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115#python-package-index-install-2002658-11>

Ook hiervoor wordt aangeraden met het testprogramma `examples/simpletest.py` te controleren of het goed functioneert.

---

---

## De MAX7219 analoge klok

---

Vanzelfsprekend is het een uitdaging om een analoge klok te presenteren op een 8×8 dot-matrix display. En omdat een 8×8 display geen centrum punt heeft hebben we gekozen voor een 7×7 presentatie.

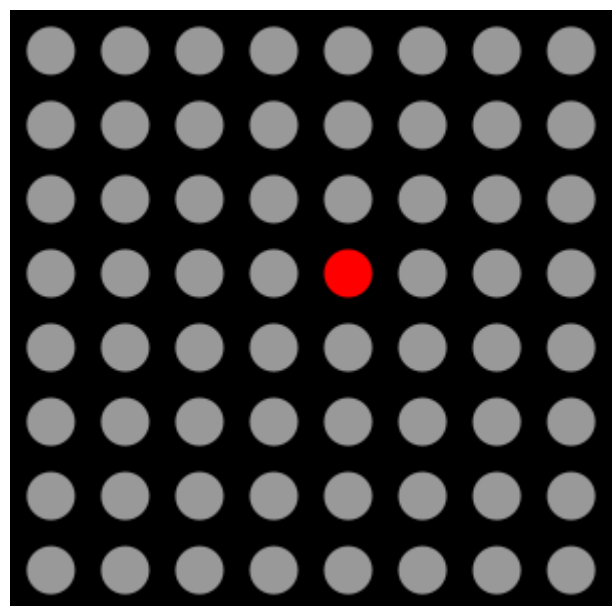
Uitgaande van positie 0,0 voor de LED links-onder hebben we een centrum punt op positie 4,4. Vanuit daar wordt er een korte lijn gemaakt voor de uren presentatie. En een lange lijn voor de minuten presentatie.

Omdat we niet elke minuut kunnen weergeven is ervoor gekozen alleen de 0,5,10,15,20 ... 55 minuten weergave te tonen.

Voor elke wijzer positie (0 t/m 12) maken we een subroutine in Python voor het weergeven van de korte en lange lijn (voorbeeld hieronder).

```
def eleven(draw, small=False):
    if small:
        draw.line((4, 4, 6, 3),
            fill="white")
    else:
        draw.line((4, 4, 7, 2),
            fill="white")
```

We maken ook een subroutine om voor elk getal (0 t/m 12) de juiste subroutine aan te roepen.



Centrum punt van de klok

```

def number(draw, n=0, small=False):
    if n == 0:
        zero(draw, small)
    if n == 1:
        one(draw, small)
    if n == 2:
        two(draw, small)
    if n == 3:
        three(draw, small)
    if n == 4:
        four(draw, small)
    if n == 5:
        five(draw, small)
    if n == 6:
        six(draw, small)
    if n == 7:
        seven(draw, small)
    if n == 8:
        eight(draw, small)
    if n == 9:
        nine(draw, small)
    if n == 10:
        ten(draw, small)
    if n == 11:
        eleven(draw, small)
    if n == 12:
        twelve(draw, small)

```

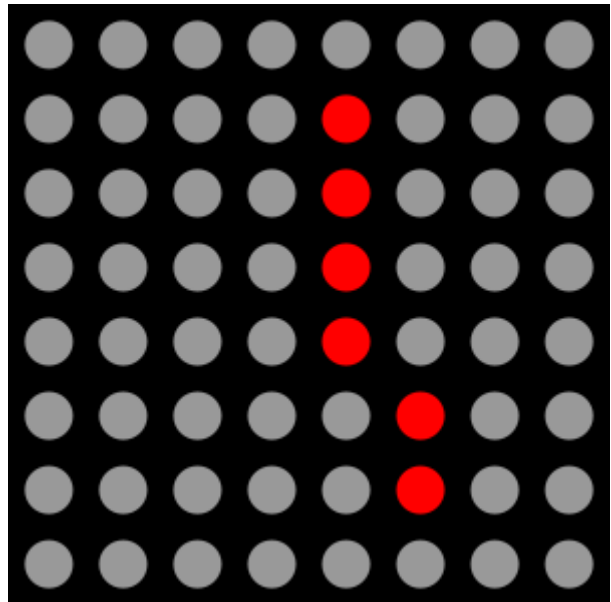
Nu we dat voor elkaar hebben kunnen we aan de hand van de tijd de juiste wijzers (lijnen) aansturen. Hiervoor zijn twee subroutines gemaakt, één voor de grote wijzer en één voor de kleine wijzer.

```

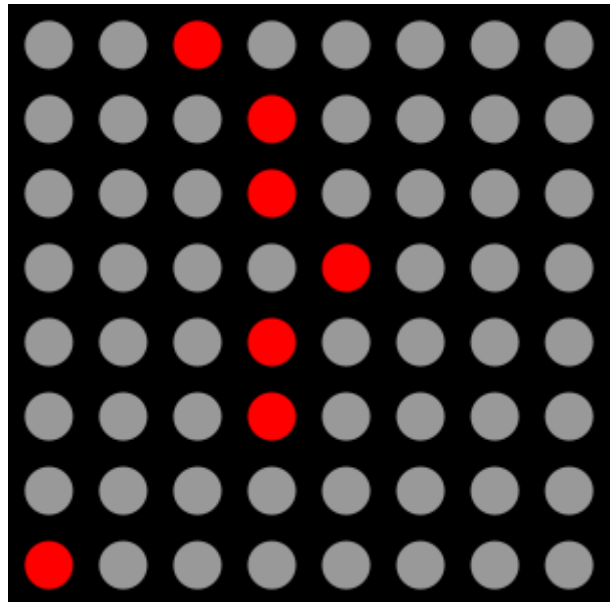
def big_hand(draw):
    now = datetime.datetime.now()
    h = now.hour
    m = now.minute
    s = now.second
    number(draw, round(m/5))
    if (s % 2) == 0:
        draw.point((0, 0), fill="black")
    if (s % 2) == 1:
        draw.point((0, 0), fill="white")

def small_hand(draw):
    now = datetime.datetime.now()
    h = now.hour
    m = now.minute
    s = now.second
    if h >= 12:
        h = h - 12
    number(draw, round(h + m/60), True)

```



Klok 5.00 uur



Klok 6.55 uur met seconde LED

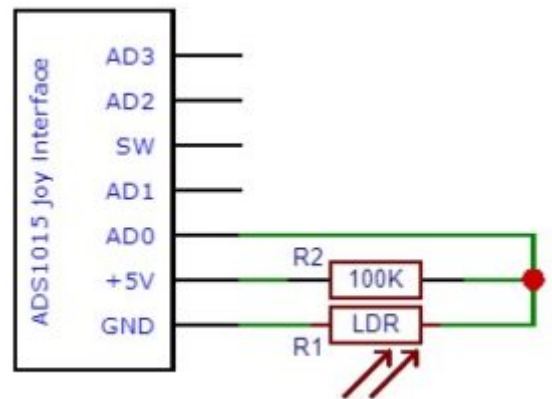
We verwerken deze code wordt daarna allemaal in de hoofdroutine (zie volledige code).

## Metten van hoeveelheid licht.

De licht hoeveelheid wordt gelezen met een LDR (licht gevoelige weerstand). Deze wordt daarvoor aangesloten op een analoge ingang module (ADS1015).

We sluiten de LDR aan tussen de GND en de analoge ingang AD0. En de 100Kohm weerstand sluiten we aan tussen de analoge ingang AD0 en de +5V. Hierdoor krijgt de analoge ingang een spanning afhankelijk van de lichtinval op de LDR.

Hoe meer licht op de LDR, hoe minder de spanning op de analoge ingang.



Zie aansluitschema rechts.

Voor het uitlezen van de lichtwaarde en aansturen van de LED display intensiteit wordt een aparte subroutine gemaakt.

```
def brightness():
    light = adc.read_adc(0, gain=1)
    if light >= 512:
        light = 512
    b = (16 - round(light/16))*16
    if b >= 256:
        b = 255
    if b < 0:
        b = 0
    device.contrast(b)
```

Deze subroutine nemen we ook in de hoofdroutine op. Daardoor wordt iedere keer de LED display intensiteit aangepast.

Het mooie van de gebruikte adafruit-ads1x15 Python Package is dat we niet zelf I<sup>2</sup>C routines hoeven te programmeren. Dat maakt het implementeren weer een stukje eenvoudiger.