

WEEKENDSCHOOL – PROGRAMMEREN –

LES 3 – BEGELEIDERSINSTRUCTIE

INTRODUCTIE

Dit is de beschrijving van de derde les programmeren. Het geeft achtergrond informatie, maar ook de uitgewerkte stappen.

LESDOELEN

Dit zijn de lesdoelen voor deze les:

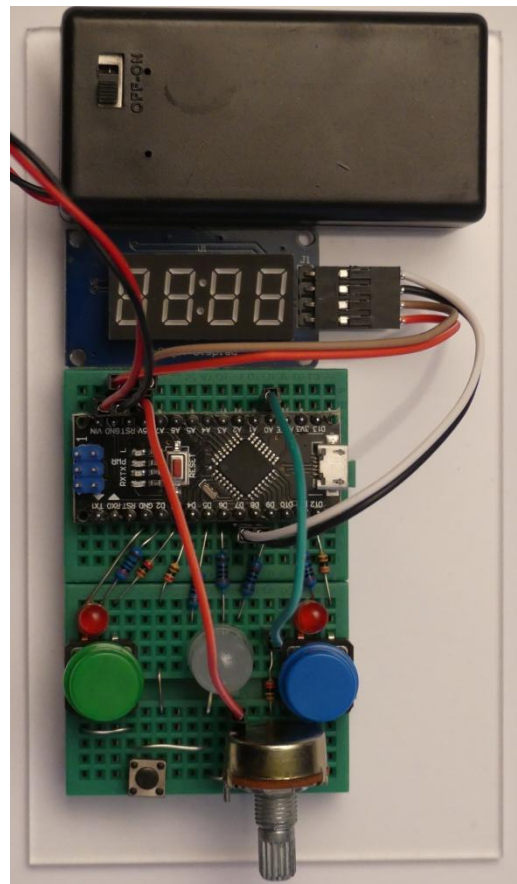
- De leerlingen laten begrijpen dat een computer ook heel klein kan zijn.
- De leerlingen laten begrijpen dat in heel veel apparatuur zo'n klein computertje zit.
- De leerlingen kennis laten maken met het programmeren op een niet grafische manier.
- De leerlingen zelf iets laten maken dat ze mee naar huis kunnen nemen.
- Maar bovenal: de leerlingen de lol van het programmeren laten zien.

WAT GAAN ZE MAKEN?

De leerlingen gaan het plankje hiernaast programmeren. Het is een spel dat uiteindelijk er om gaat dat wie het vaakst het knopje kan indrukken in een ingestelde tijd zal winnen.

Het spel start met een druk op het kleine knopje linksonder. De punten verschijnen in het display.

In de tweede helft van de les (eigenlijk de laatste ca. 45 minuten van het deel na de pauze) mogen ze zelf zo'n plankje maken. Daar is al aardig wat voorbereid, maar zij mogen de laatste hand eraan leggen. Het resultaat mogen ze mee naar huis nemen.



DE OPSTELLING

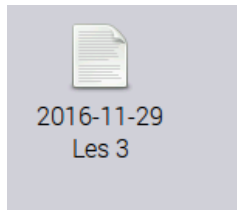
Hierbij een foto van de opstelling in de klas. Het is een foto van 4 werkplekken voor 8 leerlingen van les 2 maar geeft toch een aardige indruk.



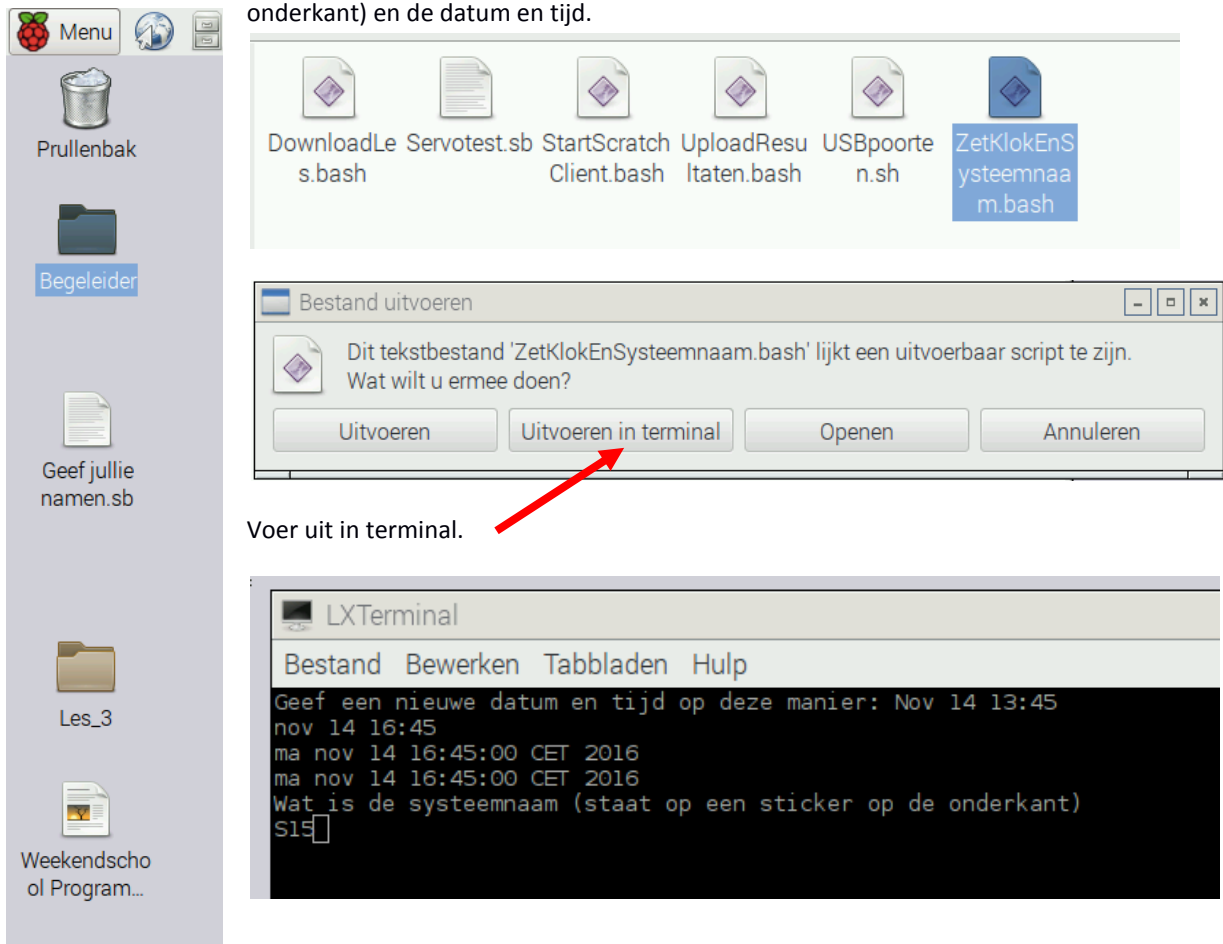
Zie het aparte document met opbouw instructies voor informatie hoe dit wordt opgezet.

DE SOFTWARE VOORBEREIDEN

Je kunt zien of iemand de software al heeft voorbereid door te kijken of je een file op het bureaublad vind met de lesdatum. Dan is het al gedaan.

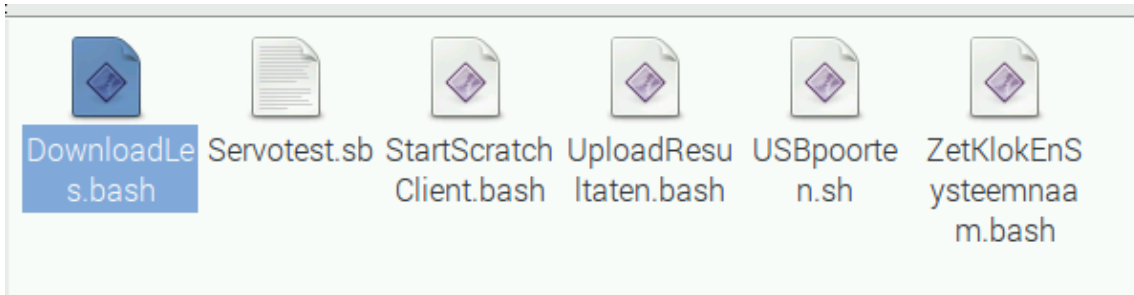


Open de map *Begeleiders* op het bureaublad. Zet dan de systeemnaam (zie sticker op de zijkant of de onderkant) en de datum en tijd.

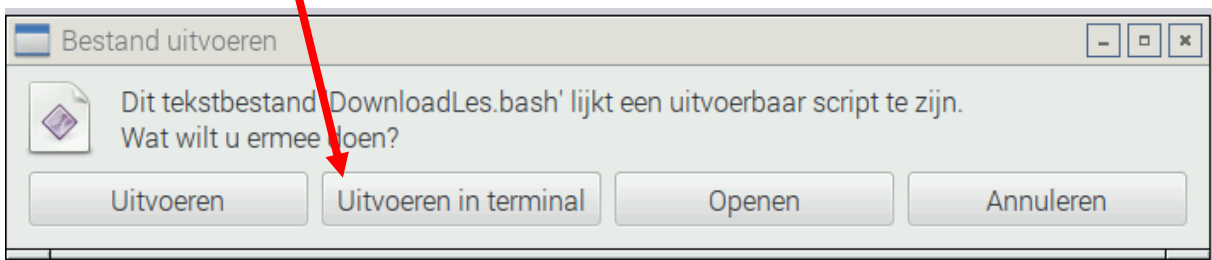


The screenshot shows the Raspberry Pi desktop environment. On the left is a sidebar with icons for 'Menu', 'Prullenbak', 'Begeleider' (highlighted), 'Geef jullie namen.sb', 'Les_3', and 'Weekendschool Program...'. The main desktop area shows several files: 'DownloadLe s.bash', 'Servotest.sb', 'StartScratch Client.bash', 'UploadResu ltaten.bash', 'USBpoorten.sh', and 'ZetKlokEnS ysteemnaa m.bash' (highlighted). A dialog box titled 'Bestand uitvoeren' is open, asking 'Dit tekstbestand 'ZetKlokEnSysteemnaam.bash' lijkt een uitvoerbaar script te zijn. Wat wilt u ermee doen?'. The buttons are 'Uitvoeren', 'Uitvoeren in terminal' (indicated by a red arrow), 'Openen', and 'Annuleren'. Below the dialog, the text 'Voer uit in terminal.' is displayed. At the bottom, an 'LXTerminal' window is open, showing the following text: 'Bestand Bewerken Tabbladen Hulp', 'Geef een nieuwe datum en tijd op deze manier: Nov 14 13:45', 'nov 14 16:45', 'ma nov 14 16:45:00 CET 2016', 'ma nov 14 16:45:00 CET 2016', 'Wat is de systeemnaam (staat op een sticker op de onderkant)', and 'S15'.

Zet de USB stick met lesmateriaal erin. Die heeft een rode sticker. Er zijn drie identieke.



Kies Uitvoeren in terminal



De software wordt nu geïnstalleerd. Als het klaar is sluit dan het venster door op *Enter* te drukken. Controleer dat er geen foutmeldingen waren.

Open Arduino en draai een test ...

< nog afmaken >

Nu de werkplek in orde is kunnen we de resterende tijd besteden om het programma door te spreken en

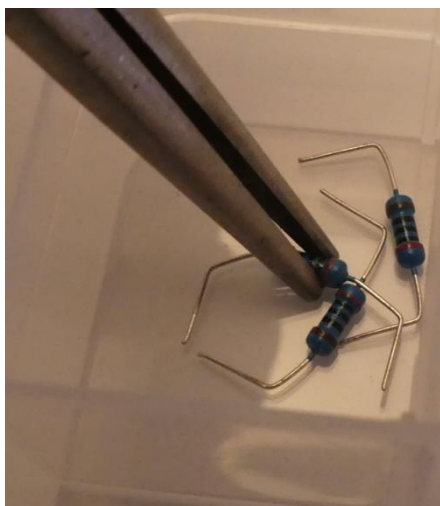
VOORBEREIDING DOOR DE BEGELEIDER: PRIEGELEN

De voorbereiding door de begeleider kost ca. 15 minuten per plankje. Er zijn voldoende bestuckte plankjes op voorraad om de les te kunnen doen, echter omdat we toch met 1 begeleider per 2 leerlingen zijn, kost het een half uur als de begeleider vóór de les het resterende voorbereidende werk doet voor twee plankjes. Dat geeft ook gelijk wat feeling met wat we in de tweede helft doen en het heeft het voordeel dat later niet iemand het 6 uur alleen moet gaan maken.

DE ONDERDELEN



Dit is het onderdelendoosje voor één leerling. Je moet er **twee** nemen om twee plankjes voor te bereiden. Om de onderdelen eruit te halen kun je het best een tang of een pincet gebruiken.



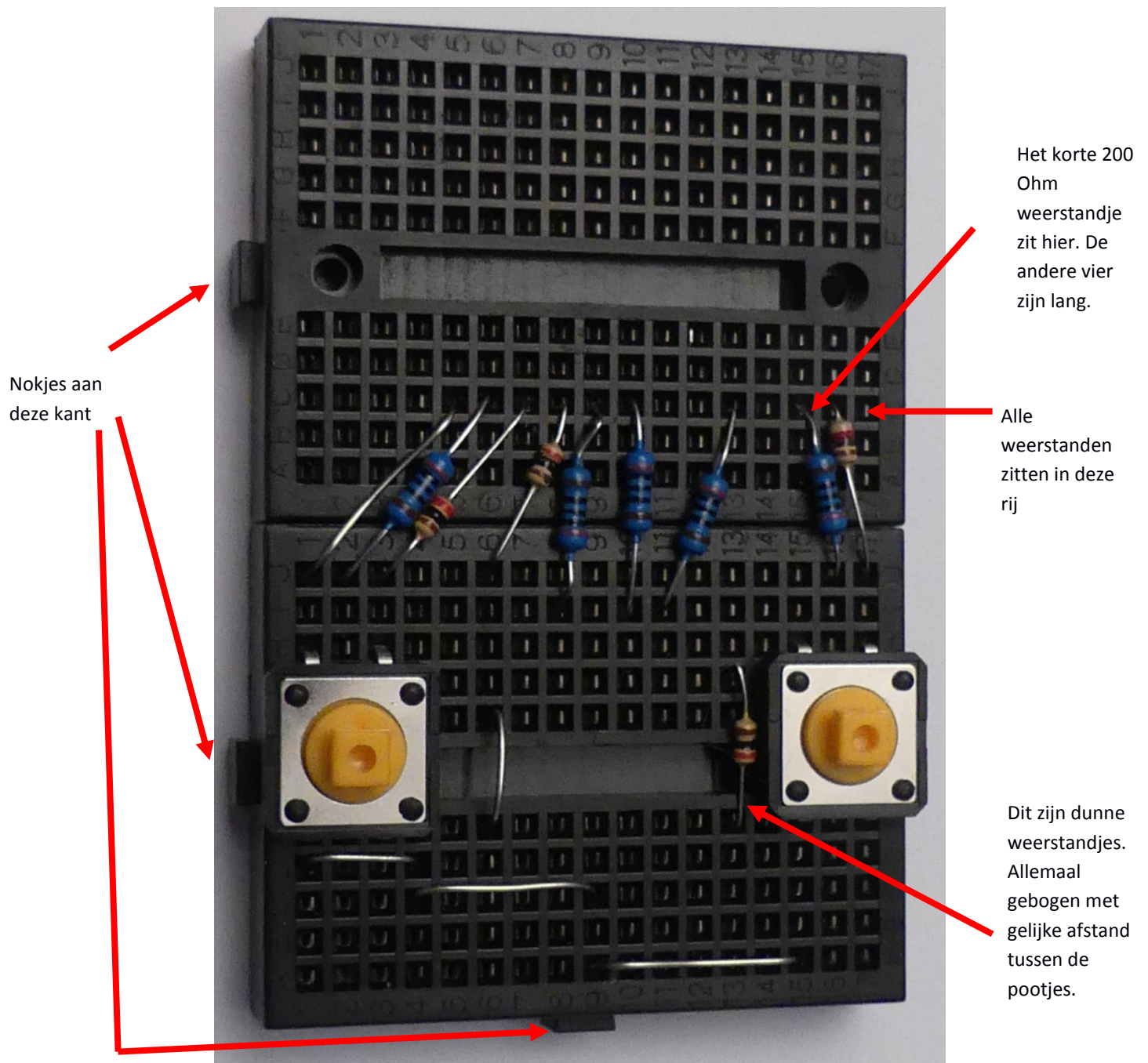
TEST VAN DE NIEUWE BREADBOARDS

- ➔ Test (per plankje) twee breadboards op klemming.

Je gebruikt een 1 kOhm weerstand die een hele dunne draad heeft. Prik het in een gaatje. Je moet een beetje weerstand voelen als je hem eruit wilt trekken. Test een aantal gaatjes over het breadboard. Als er ergens een probleem is, dan moet dat breadboard afgekeurd worden. Markeer het ding (kruis met viltstift op de achterkant) en neem een nieuwe.



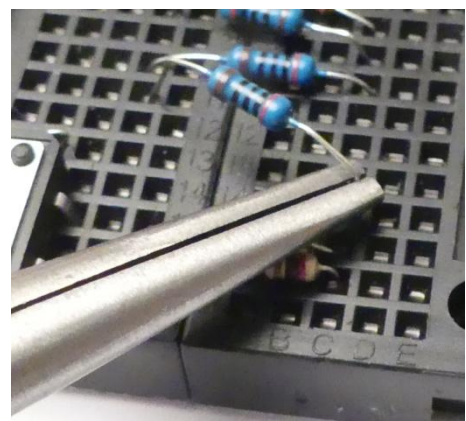
- ➔ Koppel twee breadboards aan elkaar met behulp van de nokjes. Je krijgt dan wat je ziet in de volgende foto.



- ➔ Zet alle draadbruggen en weerstanden erin.

Dat gaat het handigste door de weerstand boven de gaatjes te zetten en dan met een tangetje elke kant een klein stukje naar beneden te drukken totdat je voelt dat je door de weerstand heen bent. En daarna kun je hem met je vinger erin drukken.

- ➔ Zet ook de twee grote schakelaars erin. Pas op, want als je scheef drukt dan is er grote kans de pootjes te verbuigen.



HOE LEERLINGEN HELPEN?

Deze les probeert een balans uit tussen programmadelen te laten kopiëren en plakken en om ze zelf te laten bedenken hoe het moet. Dat is mogelijk nog niet optimaal.

In deze begeleiding staan daarom ook de uitwerkingen van de stappen. Als het helemaal niet lukt, dan kun je ze zo'n stap laten overtuiken. Maar blijf er wel bij om na te gaan dat ze snappen wat ze doen.

Zorg dat ze voortgang maken. Wees interactief, laat ze uitleggen waarom ze doen wat ze doen zodat je zeker weet dat ze het begrijpen. Als ze bijvoorbeeld nog wat willen uitproberen met de tijden van het knippen dan is dat prima, maar houd wel de voortgang erin. Als ze klaar zijn mogen ze nog naar hartenlust experimenteren.

HOEVEEL BEGELEIDERS?

Naar aanleiding van de ervaring van de vorige keer hebben we besloten dat we 1 begeleider per 2 leerlingen hebben. Dan kun je ze goed helpen en de voortgang erin houden. We wijzen de begeleiders toe aan de groepjes. Maar als er een vraag is van een ander groepje waar de begeleider even niet is, probeer die dan ook te beantwoorden.

WAAROM LEZEN ZE NIET?

Vaak zul je je afvragen: waarom lezen ze niet? Echter, in alle lessen die ik gegeven heb op de Weekendschool zien we dat lezen en doen wat er staat niet de sterkste kwaliteiten van 11-jarigen zijn.

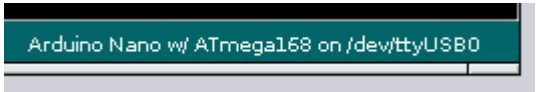
Voor een deel zit je er als begeleider zeker bij deze les bij om ze te laten lezen of ze gewoon uit te leggen wat er staat in andere woorden. Belangrijk is om ze jou te laten uitleggen wat het programma doet. Het gaat er niet om dat ze kunnen kopiëren / plakken, maar dat ze begrijpen wat ze doen. Maar om ze alles zelf te laten bedenken en intikken is ook geen optie gezien de tijd. En het is ook niet erg motiverend als je al niet gegrepen bent door programmeren. Dat moeten we dus eerst zien te bereiken.

OPMERKINGEN PER LESONDERDEEL

UITTESTEN

Als er iets anders staat dan

dan moet je in Arduino gaan naar



Arduino Nano w/ ATmega168 on /dev/ttyUSB0

Extra → Board → Arduino Nano w/ ATmega168

en

Extra → Seriele poort → /dev/ttyUSB0

NU BREIDEN WE UIT EN ... ZOEKEN WE DE FOUT!

Oplossing: er mist een punt-komma achter de regel

```
delay(500)
```

JULLIE EERSTE EIGEN PROGRAMMAREGELS

Dit zijn de regels die ze moeten toevoegen om de middelste en de rechter LED te laten branden.

```
digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
delay(500);                          // Wacht 500 milliseconde (ms) = 0,5 seconde

digitalWrite(LEDMiddenRood, LOW);    // Doe de LED weer uit
delay(100);                          // Wacht 100 milliseconde (ms) = 0,1 seconde

digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
delay(500);                          // Wacht 500 milliseconde (ms) = 0,5 seconde

digitalWrite(LEDMiddenBlauw, LOW);   // Doe de LED weer uit
delay(100);                          // Wacht 100 milliseconde (ms) = 0,1 seconde

digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
delay(500);                          // Wacht 500 milliseconde (ms) = 0,5 seconde

digitalWrite(LEDMiddenGroen, LOW);   // Doe de LED weer uit
delay(100);                          // Wacht 100 milliseconde (ms) = 0,1 seconde

digitalWrite(LEDMiddenRood, HIGH);   // Laat de middelste LED rood branden
digitalWrite(LEDMiddenBlauw, HIGH);  // Laat de middelste LED groen branden
digitalWrite(LEDMiddenGroen, HIGH);  // Laat de middelste LED groen branden
delay(500);                          // Wacht 500 milliseconde (ms) = 0,5 seconde

digitalWrite(LEDMiddenRood, LOW);    // Doe de LED weer uit
digitalWrite(LEDMiddenBlauw, LOW);   // Doe de LED weer uit
digitalWrite(LEDMiddenGroen, LOW);   // Doe de LED weer uit
delay(100);                          // Wacht 100 milliseconde (ms) = 0,1 seconde

digitalWrite(LEDRechts, HIGH);       // Laat de rechter LED branden
delay(100);                          // Wacht 100 milliseconde (ms) = 0,1 seconde

digitalWrite(LEDRechts, LOW);        // Doe de rechter LED weer uit
delay(200);                          // Wacht 200 milliseconde (ms) = 0,2 seconde
```

HOE WORDT DE PUNTENTELLING VERTOOND?

Dit is het programma op dit moment. Met geel is aangegeven waar de regels erbij kwamen. Zie hieronder voor toelichting

```
#include <TM1637Display.h>
// ==== De globale declaraties (gelden voor het hele programma) =====

// Pennen van de Arduino. We geven ze een naam zodat we in het programma een naam kunnen
// gebruiken (dat is makkelijker) en omdat we dan, als we een wijziging moeten maken
// (bijvoorbeeld een knop op een andere pin aansluiten), slechts op 1 plek een wijziging
// hoeven te maken.

// Knoppen
#define KnopLinks 3           // De linker knop zit op pen 3
#define KnopRechts 12        // De rechter knop zit op pen 12
#define KnopKlein 4          // De kleine knop linksonder zit op pen 4

// LED's
#define LEDLinks 2            // De linker LED is aangesloten op pen 2
#define LEDRechts 11         // De rechter LED is aangesloten op pen 11

// In het huisje van de middelste LED zitten eigenlijk 3 LED's: een rode, een groene en
// een blauwe. Dus deze LED is met 3 pennen aangesloten.
#define LEDMiddenRood 5       // De rode LED in de middelste (grote) LED
                                // zit op pen 5
#define LEDMiddenGroen 6      // De groene LED in de middelste (grote) LED
                                // zit op pen 9
#define LEDMiddenBlauw 9      // De blauwe LED in de middelste (grote) LED
                                // zit op pen 9

#define LEDNano 13            // De kleine LED op de Arduino Nano zelf

// Analoge ingangen
#define PotMeter A0           // De potentiometer (potmeter) is een regelbare
                                // weerstand. Die is aangesloten op pen A0

// De 4 cijfers worden bestuurd door een chip TM1637 die aan de onderkant zit
// (je kunt het transparante plaatje omdraaien en dan zie je de chip met het nummer.
// Je ziet dan ook dat er 4 pennen zijn: GND, VCC, CLK en DIO.
// VCC en GND zijn de draden voor de voeding.
// Die gaan naar resp. +5 volt en GND (= 0 volt) op de Arduino zodat de chip en het
// blok cijfers spanning krijgen.
// DIO staat voor Digitale Input / Output. Over deze draad gaat de data die vertelt welke
// cijfers moeten oplichten.
// CLK staat voor klok (clock in het Engels). Die vertelt wanneer de TM1637 chip op de
// DIO draad de data moet bekijken.
#define CLK 7                 // Dit is de klok
#define DIO 8                 // Dit is de digitale input / output

TM1637Display VierCijfers(CLK, DIO);

// ==== setup =====
// Wat tussen de accolades achter setup staat wordt (eenmaal) uitgevoerd nadat de
// Arduino is gestart.

void setup(){
    pinMode(KnopLinks, INPUT_PULLUP); // Dit is openingsaccolade die bij setup hoort.
                                        // De pen waaraan de linker knop zit is voor invoer
                                        // (input).
                                        // En er moet een optrekweerstand (pullup) worden
                                        // geconfigureerd om te zorgen dat
                                        // er een 5 volt signaal (HIGH) op de pen staat als de
                                        // knop NIET wordt ingedrukt.
                                        // Als er op de knop wordt gedrukt komt er 0 volt (LOW)
                                        // op.
    pinMode(KnopRechts, INPUT_PULLUP); // Idem voor de rechter knop
    pinMode(KnopKlein, INPUT_PULLUP);  // Idem voor de kleine knop links onder

    pinMode(LEDLinks, OUTPUT);          // De pen waaraan de linker LED zit is een
                                        // uitvoer (output).
    pinMode(LEDRechts, OUTPUT);         // Idem voor de rechter LED.
    pinMode(LEDMiddenRood, OUTPUT);     // Idem voor het rode deel van de middelste LED.
    pinMode(LEDMiddenGroen, OUTPUT);    // Idem voor het groene deel van de middelste LED.
```

```

pinMode(LEDMiddenBlauw, OUTPUT);    // Idem voor het blauwe deel van de middelste LED.

Serial.begin(115200);                // Deze opdracht is nodig om straks te kunnen zien op
                                    // de Raspberry Pi wat de Arduino aan het doen is.
                                    // 115200 wil zeggen dat de communicatie zal gaan met
                                    // 115200 bit/seconde.
                                    // Hoe je op de Raspberry Pi kunt zien wat de Arduino
                                    // doet komt later.
}                                    // Dit is de sluitaccolade die bij setup() hoort.
                                    // Dit sluit de rij opdrachten af die bij
                                    // het starten van de Arduino worden uitgevoerd.

// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop(){
    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                    // de start van de Arduino kan heel
                                    // groot worden. Dat past niet in een
                                    // int. Daar is de maximum waarde 32767.
                                    // Daarom nemen we een long.
                                    // En het getal zal nooit negatief worden,
                                    // daarom nemen we unsigned (zonder +/- teken,
                                    // dus alleen maar positieve getallen.
    unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde

    digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);    // Laat de rechter LED branden
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);     // Doe de rechter LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    VierCijfers.setBrightness(15);    // Zet de helderheid op 15. 8 = zwak,
                                    // 15 = fel
    VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
                                    // met het eerste cijfer
                                    // helemaal links (0)
    delay(1000);

```

```

VierCijfers.showNumberDec (56, false, 2, 0);    // Laat het getal 56 zien met 2 cijfers,
// eerste cijfer helemaal links
delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2);    // Laat het getal 78 zien met 2 cijfers,
// start vanaf het derde cijfer (plek 2,
// want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0};      // declareer een rij van 2 cijfers en
// maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0);  // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2);  // Wis de rechter 2 cijfers
}

```

TOELICHTING

Met `#include` zorgen we dat de referenties naar de bibliotheek opgelost kunnen worden. Dus dat het object `TM1637Display` kan worden gemaakt met de bijbehorende methoden.

De regel

```
TM1637Display VierCijfers (CLK, DIO);
```

maakt het object `VierCijfers` van het type `TM1637Display`.

De regel

```
const uint8_t TweeCijfersLeeg[] = {0, 0};
```

definieert een array met twee elementen van een byte (`uint8_t` is een unsigned integer van 8 bits, dus 1 byte).

De bits in de getallen geven aan welke segmenten van het 7-segment display aan moeten. Alles is 0, dus geen enkel segment moet dan oplichten.

De regel

```
VierCijfers.setSegments(TweeCijfersLeeg,2,0);
```

zet de segmenten die moeten oplichten. Er wordt een array van 2 cijfers meegegeven en dan specificeren we dat er 2 cijfers moeten worden veranderd startend vanaf plek 0, het meest linkse cijfer.

We hadden ook een array van 4 nullen kunnen maken en dan alle 4 cijfers met 1 opdracht kunnen wissen, echter we hebben straks nodig dat we het per 2 cijfers kunnen doen en dan hebben ze dit al gezien. Daarom doen we het hier per 2 cijfers.

DE KNOPPEN EN VARIABLEN

Dit is het aangepaste programma, met in geel de aangepaste regels. Het is alleen het deel van loop(). De rest is ongewijzigd.

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop() {

  int StatusKnopKleinNu;
  unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.
  unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde

  digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
  delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
  delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

  digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
  delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

  digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
  delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
  delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

  digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
  delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
  delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

  digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
  delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
  digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
  digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
  delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

  digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
  digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
  digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
  delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDRechts, HIGH);    // Laat de rechter LED branden
  delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

  digitalWrite(LEDRechts, LOW);     // Doe de rechter LED weer uit
  delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

  VierCijfers.setBrightness(15);     // Zet de helderheid op 15. 8 = zwak,
                                     // 15 = fel
  VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
                                                  // met het eerste cijfer
                                                  // helemaal links (0)

  delay(1000);
```



```

VierCijfers.showNumberDec (56, false, 2, 0); // Laat het getal 56 zien met 2 cijfers,
// eerste cijfer helemaal links
delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2); // Laat het getal 78 zien met 2 cijfers,
// start vanaf het derde cijfer (plek 2,
// want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0}; // declareer een rij van 2 cijfers en
// maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

```

```

StatusKnopKleinNu = digitalRead(KnopKlein);

```

```

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

```

DE OPERATOREN VOOR VERGELIJKING

Bedenk dat je ==, && en || moet gebruiken in een test. Als je bijvoorbeeld zou schrijven if (a = 0), dan wordt a 0 gemaakt. Een & is een bitwise AND en | is een bitwise OR. In vergelijkingen daarom altijd de dubbele tekens gebruiken.

DE COMPACTERE VORM

Je kunt hetzelfde bereiken met een enkele opdracht

```

digitalWrite(LEDNano,!digitalRead(KnopKlein));

```

De opdracht digitalRead geeft LOW of HIGH. De opdracht digitalWrite gebruikt dezelfde waarden. Echter als de knop ingedrukt is, is de waarde LOW, terwijl we dan de LED de waarde HIGH willen geven.

Omdat LOW en HIGH eigenlijk 0 en 1 zijn, net zoals true en false, kunnen we de NOT operator gebruiken. Dat is in C een uitroepteken. Daarmee keren we de waarden om. 0 wordt 1 en 1 wordt 0.

EERST VOOR DE LINKER SPELER

Dit is wat de code in loop() nu is, met in geel de toevoegingen.

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop() {

    int StatusKnopKleinNu;
    int StatusKnopLinksNu;
    int StatusKnopLinksLaatst;
    int PuntenLinks;

    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.

    unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde


    digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
    delay(200);                       // Wacht 200 milliseconde (ms) = 0,2 seconde


    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);    // Laat de rechter LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);     // Doe de rechter LED weer uit
    delay(200);                       // Wacht 200 milliseconde (ms) = 0,2 seconde


    VierCijfers.setBrightness(15);    // Zet de helderheid op 15. 8 = zwak,
                                     // 15 = fel
    VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
                                     // met het eerste cijfer
}
```

```

// helemaal links (0)
delay(1000);
VierCijfers.showNumberDec (56, false, 2, 0); // Laat het getal 56 zien met 2 cijfers,
// eerste cijfer helemaal links

delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2); // Laat het getal 78 zien met 2 cijfers,
// start vanaf het derde cijfer (plek 2,
// want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0}; // declareer een rij van 2 cijfers en
// maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

StatusKnopKleinNu = digitalRead(KnopKlein);

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

// ==== Hier begint het spel =====
while (true) { // Herhaal dit eendeloos
    if (digitalRead(KnopKlein) == LOW) { // Als het kleine knopje ingedrukt wordt
        // dan gaan we van start

        // ==== PLAATS 3 begin ==== Na deze regel komt straks het instellen van de speeltijd ==
        // ==== PLAATS 3 eind ===== Voor deze regel komt straks het instellen van de speeltijd =

        // ==== Voorbereidingen voor het spel =====

        StartTijd = millis(); // Onthoud hoe laat het nu is
        PuntenLinks = 0; // Zet de punten van de linker speler op
        // nul
        VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
        // speler in de twee linker cijfers
        StatusKnopLinksNu = digitalRead(KnopLinks); // Lees de status van de linker knop
        StatusKnopLinksLaatst = StatusKnopLinksNu; // En dat is dan ook de beginwaarde

        while ((millis() < StartTijd + Speeltijd * 1000) && (PuntenLinks < 99) ) {
            // Als de huidige tijd (millis()) kleiner
            // is dan de starttijd plus de speeltijd
            // (die we met 1000 vermenigvuldigen om
            // ook milliseconde te krijgen), dan
            // kunnen we doorgaan: de speeltijd is
            // nog niet om.
            // Maar als de speler al 99 punten heeft
            // dan moeten we toch stoppen, want
            // meer dan 99 punten kunnen we niet
            // laten zien.

            // ==== PLAATS 1 begin ==== Hieronder komen straks regels voor de knoppen =====
            StatusKnopLinksNu = digitalRead(KnopLinks);
            if (StatusKnopLinksNu == LOW && StatusKnopLinksLaatst == HIGH) {
                // Als de knop nu LOW is en hij was HIGH
                // in de vorige keer dat we door de lus
                // kwamen, dan is de knop nu net ingedrukt
                PuntenLinks = PuntenLinks + 1; // De linker speler krijgt er een punt bij
                VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
                // speler in de twee linker cijfers
            }
            StatusKnopLinksLaatst = StatusKnopLinksNu; // Onthoud de waarde voor de volgende keer
            // dat we door de lus komen

            // ==== PLAATS 1 eind ===== Voor deze regel komen de opdrachten voor de knoppen =====
        }

        // ==== PLAATS 2 begin ==== Hieronder komen straks de regels voor de winnaar =====
        // ==== PLAATS 2 eind ===== Hierboven komen straks de regels voor de winnaar =====
    }
}

```

```
}  
}
```

NU VOOR DE RECHTER SPELER

Hieronder staat het programma met de toevoegingen voor de rechter speler in geel.

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop() {

    int StatusKnopKleinNu;
    int StatusKnopLinksNu;
    int StatusKnopLinksLaatst;
    int PuntenLinks;

    int StatusKnopRechtsNu;
    int StatusKnopRechtsLaatst;
    int PuntenRechts;

    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.

    unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde


    digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
    delay(200);                       // Wacht 200 milliseconde (ms) = 0,2 seconde


    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);    // Laat de rechter LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);     // Doe de rechter LED weer uit
    delay(200);                       // Wacht 200 milliseconde (ms) = 0,2 seconde
```



```

VierCijfers.setBrightness(15); // Zet de helderheid op 15. 8 = zwak,
                                // 15 = fel
VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
                                                // met het eerste cijfer
                                                // helemaal links (0)

delay(1000);
VierCijfers.showNumberDec (56, false, 2, 0); // Laat het getal 56 zien met 2 cijfers,
                                                // eerste cijfer helemaal links

delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2); // Laat het getal 78 zien met 2 cijfers,
                                                // start vanaf het derde cijfer (plek 2,
                                                // want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0}; // declareer een rij van 2 cijfers en
                                                // maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

StatusKnopKleinNu = digitalRead(KnopKlein);

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

// ==== Hier begint het spel =====
while (true) { // Herhaal dit eindeloos
    if (digitalRead(KnopKlein) == LOW) { // Als het kleine knopje ingedrukt wordt
                                        // dan gaan we van start

        // ==== PLAATS 3 begin ==== Na deze regel komt straks het instellen van de speeltijd ==
        // ==== PLAATS 3 eind ===== Voor deze regel komt straks het instellen van de speeltijd =

        // ==== Voorbereidingen voor het spel =====

        StartTijd = millis(); // Onthoud hoe laat het nu is
        PuntenLinks = 0; // Zet de punten van de linker speler op
                            // nul
        VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
                                                                // speler in de twee linker cijfers
        PuntenRechts = 0; // Zet de punten van de linker speler op
                            // nul
        VierCijfers.showNumberDec(PuntenRechts, false, 2, 2); // Vertoon het aantal punten van de
                                                                // rechter
                                                                // speler in de twee rechter cijfers
        StatusKnopLinksNu = digitalRead(KnopLinks); // Lees de status van de linker knop
        StatusKnopLinksLaatst = StatusKnopLinksNu; // En dat is dan ook de beginwaarde

        StatusKnopRechtsNu = digitalRead(KnopRechts); // Lees de status van de linker knop
        StatusKnopRechtsLaatst = StatusKnopRechtsNu; // En dat is dan ook de beginwaarde

        while ((millis() < StartTijd + SpeelTijd * 1000) && (PuntenLinks < 99) && (PuntenRechts
< 99)) {
            // Als de huidige tijd (millis()) kleiner
            // is dan de starttijd plus de speeltijd
            // (die we met 1000 vermenigvuldigen om
            // ook milliseconde te krijgen), dan
            // kunnen we doorgaan: de speeltijd is
            // nog niet om.
            // Maar als de speler al 99 punten heeft
            // dan moeten we toch stoppen, want
            // meer dan 99 punten kunnen we niet
            // laten zien.

            // ==== PLAATS 1 begin ==== Hieronder komen straks regels voor de knoppen =====
            StatusKnopLinksNu = digitalRead(KnopLinks);
            if (StatusKnopLinksNu == LOW && StatusKnopLinksLaatst == HIGH) {
                // Als de knop nu LOW is en hij was HIGH
                // in de vorige keer dat we door de lus
                // kwamen, dan is de knop nu net ingedrukt
                PuntenLinks = PuntenLinks + 1; // De linker speler krijgt er een punt bij
                VierCijfers.showNumberDec(PuntenLinks, false, 2, 0);
            }
        }
    }
}

```

```

// Vertoon het aantal punten van de linker
// speler in de twee linker cijfers
}
StatusKnopLinksLaatst = StatusKnopLinksNu; // Onthoud de waarde voor de volgende keer
// dat we door de lus komen

StatusKnopRechtsNu = digitalRead(KnopRechts);
if (StatusKnopRechtsNu == LOW && StatusKnopRechtsLaatst == HIGH) {
    // Als de knop nu LOW is en hij was HIGH
    // in de vorige keer dat we door de lus
    // kwamen, dan is de knop nu net ingedrukt
    PuntenRechts = PuntenRechts + 1; // De rechter speler krijgt er een puntbij
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    // Vertoon het aantal punten van de
    // rechter
    // speler in de twee rechter cijfers
}
StatusKnopRechtsLaatst = StatusKnopRechtsNu; // Onthoud de waarde voor de volgende
// keer dat we door de lus komen

// ==== PLAATS 1 eind ===== Voor deze regel komen de opdrachten voor de knoppen =====
}

// ==== PLAATS 2 begin ===== Hieronder komen straks de regels voor de winnaar =====
// ==== PLAATS 2 eind ===== Hierboven komen straks de regels voor de winnaar =====
}
}
}

```

WIE HEEFT ER GEWONNEN?

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop() {

    int StatusKnopKleinNu;
    int StatusKnopLinksNu;
    int StatusKnopLinksLaatst;
    int PuntenLinks;

    int StatusKnopRechtsNu;
    int StatusKnopRechtsLaatst;
    int PuntenRechts;

    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.
    unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde

    // =====

    digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    // =====

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                        // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH);  // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                       // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);   // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);     // Laat de rechter LED branden
    delay(100);                       // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);      // Doe de rechter LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    // =====

    VierCijfers.setBrightness(15);      // Zet de helderheid op 15. 8 = zwak,
                                     // 15 = fel
    VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
}
```

```

// met het eerste cijfer
// helemaal links (0)

delay(1000);
VierCijfers.showNumberDec (56, false, 2, 0); // Laat het getal 56 zien met 2 cijfers,
// eerste cijfer helemaal links

delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2); // Laat het getal 78 zien met 2 cijfers,
// start vanaf het derde cijfer (plek 2,
// want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0}; // declareer een rij van 2 cijfers en
// maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

StatusKnopKleinNu = digitalRead(KnopKlein);

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

// ==== Hier begint het spel =====
while (true) { // Herhaal dit eindeloos
    if (digitalRead(KnopKlein) == LOW) { // Als het kleine knopje ingedrukt wordt
// dan gaan we van start

// ==== PLAATS 3 begin ==== Na deze regel komt straks het instellen van de speeltijd ==
// ==== PLAATS 3 eind ===== Voor deze regel komt straks het instellen van de speeltijd =

// ==== Voorbereidingen voor het spel =====

StartTijd = millis(); // Onthoud hoe laat het nu is
PuntenLinks = 0; // Zet de punten van de linker speler op
// nul
VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
// speler in de twee linker cijfers
PuntenRechts = 0; // Zet de punten van de linker speler op
// nul
VierCijfers.showNumberDec(PuntenRechts, false, 2, 2); // Vertoon het aantal punten van de
// rechter
// speler in de twee rechter cijfers
StatusKnopLinksNu = digitalRead(KnopLinks); // Lees de status van de linker knop
StatusKnopLinksLaatst = StatusKnopLinksNu; // En dat is dan ook de beginwaarde

StatusKnopRechtsNu = digitalRead(KnopRechts); // Lees de status van de linker knop
StatusKnopRechtsLaatst = StatusKnopRechtsNu; // En dat is dan ook de beginwaarde

while ((millis() < StartTijd + SpeelTijd * 1000) && (PuntenLinks < 99) && (PuntenRechts
< 99) ) {
// Als de huidige tijd (millis()) kleiner
// is dan de starttijd plus de speeltijd
// (die we met 1000 vermenigvuldigen om
// ook milliseconde te krijgen), dan
// kunnen we doorgaan: de speeltijd is
// nog niet om.
// Maar als de speler al 99 punten heeft
// dan moeten we toch stoppen, want
// meer dan 99 punten kunnen we niet
// laten zien.

// ==== PLAATS 1 begin ==== Hieronder komen straks regels voor de knoppen =====
StatusKnopLinksNu = digitalRead(KnopLinks);
if (StatusKnopLinksNu == LOW && StatusKnopLinksLaatst == HIGH) {
// Als de knop nu LOW is en hij was HIGH
// in de vorige keer dat we door de lus
// kwamen, dan is de knop nu net ingedrukt
PuntenLinks = PuntenLinks + 1; // De linker speler krijgt er een punt bij
VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
// speler in de twee linker cijfers
}
}

```

```

StatusKnopLinksLaatst = StatusKnopLinksNu; // Onthoud de waarde voor de volgende keer
// dat we door de lus komen

StatusKnopRechtsNu = digitalRead(KnopRechts);
if (StatusKnopRechtsNu == LOW && StatusKnopRechtsLaatst == HIGH) {
    // Als de knop nu LOW is en hij was HIGH
    // in de vorige keer dat we door de lus
    // kwamen, dan is de knop nu net ingedrukt
    PuntenRechts = PuntenRechts + 1; // De rechter speler krijgt er een puntbij
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    // Vertoon het aantal punten van de
    // rechter
    // speler in de twee rechter cijfers
}
StatusKnopRechtsLaatst = StatusKnopRechtsNu; // Onthoud de waarde voor de volgende
// keer dat we door de lus komen

// ==== PLAATS 1 eind ==== Voor deze regel komen de opdrachten voor de knoppen =====
}

// ==== PLAATS 2 begin ==== Hieronder komen straks de regels voor de winnaar =====
// Deze ronde is afgelopen. Laat nu de punten van de winnaar, of van beide bij gelijk
// spel knipperen. Dat doen we door eerst de cijfers van de winnaar (of beiden) weg te
// halen en dan beide weer te laten zien.
while (digitalRead(KnopKlein) == HIGH) {
    // Als de kleine knop wordt
    // ingedrukt dan stoppen we met
    // te laten zien wie heeft
    // gewonnen.

    if (PuntenLinks >= PuntenRechts) {
        // Als Links heeft gewonnen of als
        // het gelijk is ...
        VierCijfers.setSegments(TweeCijfersLeeg, 2, 0); // Wis de linker 2 cijfers
    }

    if (PuntenRechts >= PuntenLinks) {
        VierCijfers.setSegments(TweeCijfersLeeg, 2, 2); // Wis de rechter 2 cijfers
    }

    delay (100);
    VierCijfers.showNumberDec(PuntenLinks, false, 2, 0);
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    delay (100);
}

// ==== PLAATS 2 eind ==== Hierboven komen straks de regels voor de winnaar =====
}
}
}

```


SPEELTIJD INSTELLEN

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop(){

    int StatusKnopKleinNu;
    int StatusKnopLinksNu;
    int StatusKnopLinksLaatst;
    int PuntenLinks;

    int StatusKnopRechtsNu;
    int StatusKnopRechtsLaatst;
    int PuntenRechts;

    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.
    unsigned long SpeelTijd = 15;     // de speeltijd is 15 seconde

    digitalWrite(LEDLinks, HIGH);    // Laat de linker LED branden
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);     // Doe de linker LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    delay(100);                     // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    delay(100);                     // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                     // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                      // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW); // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW); // Doe de LED weer uit
    delay(100);                     // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);    // Laat de rechter LED branden
    delay(100);                      // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);     // Doe de rechter LED weer uit
    delay(200);                      // Wacht 200 milliseconde (ms) = 0,2 seconde

    VierCijfers.setBrightness(15);    // Zet de helderheid op 15. 8 = zwak,
                                     // 15 = fel
}
```

```

VierCijfers.showNumberDec (1234, false, 4, 0);    // Laat het getal 1234 zien op 4 plekken
                                                    // met het eerste cijfer
                                                    // helemaal links (0)

delay(1000);
VierCijfers.showNumberDec (56, false, 2, 0);      // Laat het getal 56 zien met 2 cijfers,
                                                    // eerste cijfer helemaal links

delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2);      // Laat het getal 78 zien met 2 cijfers,
                                                    // start vanaf het derde cijfer (plek 2,
                                                    // want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0};        // declareer een rij van 2 cijfers en
                                                    // maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0);    // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2);    // Wis de rechter 2 cijfers

StatusKnopKleinNu = digitalRead(KnopKlein);

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

// ==== Hier begint het spel =====
while (true) {                                    // Herhaal dit eindeloos
    if (digitalRead(KnopKlein) == LOW) {          // Als het kleine knopje ingedrukt wordt
                                                    // dan gaan we van start

        // ==== PLAATS 3 begin ==== Na deze regel komt straks het instellen van de speeltijd ===
        // ==== Stel nu eerst de speeltijd in
        VierCijfers.setSegments(TweeCijfersLeeg,2,0);    // Wis de linker 2 cijfers
        VierCijfers.setSegments(TweeCijfersLeeg,2,2);    // Wis de rechter 2 cijfers

        int i;
        int PotMeterStand;
        for (i=1; i<1000; i++) {                  // doe het 1000 keer om de spelers
                                                    // een beetje tijd te geven
            PotMeterStand = analogRead(PotMeter);      // lees de stand van de potmeter

            SpeelTijd = map(PotMeterStand,0,1024,5,60); // vertaal het naar een bruikbare
                                                    // waarde voor de speeltijd
            VierCijfers.showNumberDec(SpeelTijd, false,2, 1); // laat de speeltijd zien

            // ==== PLAATS 4 begin ==== Hieronder komen straks de regels voor de toegift =====
            // ==== PLAATS 4 eind ===== Hierboven komen straks de regels voor de toegift =====
        }

        // ==== PLAATS 3 eind ===== Voor deze regel komt straks het instellen van de speeltijd =

        // ==== Voorbereidingen voor het spel =====

        StartTijd = millis();                      // Onthoud hoe laat het nu is
        PuntenLinks = 0;                           // Zet de punten van de linker speler op
                                                    // nul
        VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
                                                    // speler in de twee linker cijfers
        PuntenRechts = 0;                           // Zet de punten van de linker speler op
                                                    // nul
        VierCijfers.showNumberDec(PuntenRechts, false, 2, 2); // Vertoon het aantal punten van de
                                                    // rechter
                                                    // speler in de twee rechter cijfers
        StatusKnopLinksNu = digitalRead(KnopLinks); // Lees de status van de linker knop
        StatusKnopLinksLaatst = StatusKnopLinksNu; // En dat is dan ook de beginwaarde

        StatusKnopRechtsNu = digitalRead(KnopRechts); // Lees de status van de linker knop
        StatusKnopRechtsLaatst = StatusKnopRechtsNu; // En dat is dan ook de beginwaarde

        while ((millis() < StartTijd + SpeelTijd * 1000) && (PuntenLinks < 99) && (PuntenRechts
< 99) ) {
                                                    // Als de huidige tijd (millis()) kleiner

```

```

// is dan de starttijd plus de speeltijd
// (die we met 1000 vermenigvuldigen om
// ook milliseconde te krijgen), dan
// kunnen we doorgaan: de speeltijd is
// nog niet om.
// Maar als de speler al 99 punten heeft
// dan moeten we toch stoppen, want
// meer dan 99 punten kunnen we niet
// laten zien.

// ==== PLAATS 1 begin ==== Hieronder komen straks regels voor de knoppen =====
StatusKnopLinksNu = digitalRead(KnopLinks);
if (StatusKnopLinksNu == LOW && StatusKnopLinksLaatst == HIGH) {
    // Als de knop nu LOW is en hij was HIGH
    // in de vorige keer dat we door de lus
    // kwamen, dan is de knop nu net ingedrukt
    PuntenLinks = PuntenLinks + 1; // De linker speler krijgt er een punt bij
    VierCijfers.showNumberDec(PuntenLinks, false, 2, 0);
    // Vertoon het aantal punten van de linker
    // speler in de twee linker cijfers
}
StatusKnopLinksLaatst = StatusKnopLinksNu; // Onthoud de waarde voor de volgende keer
// dat we door de lus komen

StatusKnopRechtsNu = digitalRead(KnopRechts);
if (StatusKnopRechtsNu == LOW && StatusKnopRechtsLaatst == HIGH) {
    // Als de knop nu LOW is en hij was HIGH
    // in de vorige keer dat we door de lus
    // kwamen, dan is de knop nu net ingedrukt
    PuntenRechts = PuntenRechts + 1; // De rechter speler krijgt er een punt
    // bij
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    // Vertoon het aantal punten van de
    // rechter
    // speler in de twee rechter cijfers
}
StatusKnopRechtsLaatst = StatusKnopRechtsNu; // Onthoud de waarde voor de volgende
// keer dat we door de lus komen

// ==== PLAATS 1 eind ===== Voor deze regel komen de opdrachten voor de knoppen =====
}

// ==== PLAATS 2 begin ==== Hieronder komen straks de regels voor de winnaar =====
// Deze ronde is afgelopen. Laat nu de punten van de winnaar, of van beide bij gelijk
// spel knipperen. Dat doen we door eerst de cijfers van de winnaar (of beiden) weg te
// halen en dan beide weer te laten zien.
while (digitalRead(KnopKlein) == HIGH) {
    // Als de kleine knop wordt
    // ingedrukt dan stoppen we met
    // te laten zien wie heeft
    // gewonnen.

    if (PuntenLinks >= PuntenRechts) {
        // Als Links heeft gewonnen of als
        // het gelijk is ...
        VierCijfers.setSegments(TweeCijfersLeeg, 2, 0); // Wis de linker 2 cijfers
    }

    if (PuntenRechts >= PuntenLinks) {
        VierCijfers.setSegments(TweeCijfersLeeg, 2, 2); // Wis de rechter 2 cijfers
    }

    delay (100);
    VierCijfers.showNumberDec(PuntenLinks, false, 2, 0);
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    delay (100);
}

// ==== PLAATS 2 eind ===== Hierboven komen straks de regels voor de winnaar =====
}
}
}

```

WERKT DE POTMETER LOGISCH?

De range van 0 tot 1024 moet vertaald worden naar een range van 60 tot 5 i.p.v. 5 tot 60. Dit is de aangepaste regel: draai de 60 en de 5 om.

```
SpeelTijd = map(PotMeterStand, 0, 1024, 60, 5);
```

KNUTSEL ZELF EEN PLANKJE IN ELKAAR – OF IETS ANDERS

Als ze hier gekomen zijn, maar uiterlijk om ca. 13.30 uur, moeten ze kiezen wat ze gaan doen:

1. Plankje maken dat ze mee naar huis mogen nemen om het spel te spelen. Ze krijgen een batterij of netvoeding met kabel mee dat is het enige dat ze nodig hebben. Ze hebben geen PC of RPi nodig.
2. Verder gaan met het programmeren van dit plankje
3. Iets programmeren in Scratch.

Als ze het nog niet af hebben en ze willen een plankje maken dan moeten ze dat nu doen, anders is er geen tijd meer. Ze kunnen straks verder programmeren en je kunt ze eventueel daarbij een extra handje helpen.

Ze moeten alleen een plankje maken om mee naar huis te nemen als ze dat echt willen en er iets mee denken te gaan doen (laten zien aan familie, op school, spelen met vriendjes etc.).

Als de helft van je groepje het een wil en de ander het andere, overleg dan even met Marga of we kinderen uit groepjes kunnen wisselen.

LICHT IN ALLE REGENBOOG KLEUREN

```
// ==== loop =====
// Wat tussen de accolades staat wordt uitgevoerd na setup(). En dat wordt eindeloos herhaald.

void loop() {

    int StatusKnopKleinNu;
    int StatusKnopLinksNu;
    int StatusKnopLinksLaatst;
    int PuntenLinks;

    int StatusKnopRechtsNu;
    int StatusKnopRechtsLaatst;
    int PuntenRechts;

    unsigned long StartTijd;          // De tijd (in milli seconde vanaf de
                                     // de start van de Arduino kan heel
                                     // groot worden. Dat past niet in een
                                     // int. Daar is de maximum waarde 32767.
                                     // Daarom nemen we een long.
                                     // En het getal zal nooit negatief worden,
                                     // daarom nemen we unsigned (zonder +/- teken,
                                     // dus alleen maar positieve getallen.
    unsigned long SpeelTijd = 15;      // de speeltijd is 15 seconde

    digitalWrite(LEDLinks, HIGH);      // Laat de linker LED branden
    delay(100);                         // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDLinks, LOW);       // Doe de linker LED weer uit
    delay(200);                         // Wacht 200 milliseconde (ms) = 0,2 seconde

    digitalWrite(LEDMiddenRood, HIGH); // Laat de middelste LED rood branden
    delay(500);                         // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);  // Doe de LED weer uit
    delay(100);                        // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    delay(500);                        // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
    delay(100);                        // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                        // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
    delay(100);                        // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDMiddenRood, HIGH);  // Laat de middelste LED rood branden
    digitalWrite(LEDMiddenBlauw, HIGH); // Laat de middelste LED groen branden
    digitalWrite(LEDMiddenGroen, HIGH); // Laat de middelste LED groen branden
    delay(500);                         // Wacht 500 milliseconde (ms) = 0,5 seconde

    digitalWrite(LEDMiddenRood, LOW);   // Doe de LED weer uit
    digitalWrite(LEDMiddenBlauw, LOW);  // Doe de LED weer uit
    digitalWrite(LEDMiddenGroen, LOW);  // Doe de LED weer uit
    delay(100);                         // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, HIGH);      // Laat de rechter LED branden
    delay(100);                         // Wacht 100 milliseconde (ms) = 0,1 seconde

    digitalWrite(LEDRechts, LOW);       // Doe de rechter LED weer uit
    delay(200);                         // Wacht 200 milliseconde (ms) = 0,2 seconde

    VierCijfers.setBrightness(15);      // Zet de helderheid op 15. 8 = zwak,
                                     // 15 = fel
    VierCijfers.showNumberDec (1234, false, 4, 0); // Laat het getal 1234 zien op 4 plekken
}
```



```

// met het eerste cijfer
// helemaal links (0)

delay(1000);
VierCijfers.showNumberDec (56, false, 2, 0); // Laat het getal 56 zien met 2 cijfers,
// eerste cijfer helemaal links

delay(1000);
VierCijfers.showNumberDec (78, false, 2, 2); // Laat het getal 78 zien met 2 cijfers,
// start vanaf het derde cijfer (plek 2,
// want het eerste cijfer is plek 0)

delay(1000);
const uint8_t TweeCijfersLeeg[] = {0, 0}; // declareer een rij van 2 cijfers en
// maak ze allemaal 0
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

StatusKnopKleinNu = digitalRead(KnopKlein);

if (StatusKnopKleinNu == LOW) {
    digitalWrite(LEDNano,HIGH);
} else {
    digitalWrite(LEDNano,LOW);
}

// ==== Hier begint het spel =====
while (true) { // Herhaal dit eindeloos
    if (digitalRead(KnopKlein) == LOW) { // Als het kleine knopje ingedrukt wordt
        // dan gaan we van start

        // ==== PLAATS 3 begin ==== Na deze regel komt straks het instellen van de speeltijd ==
        // ==== Stel nu eerst de speeltijd in
        VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
        VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

        int i;
        int PotMeterStand;
        int PotMeterStandBegin;
        int LEDWaardeBegin;
        int LEDWaardeNieuw;
        int LEDMiddenWaardeRood = 0;
        int LEDMiddenWaardeGroen = 0;
        int LEDMiddenWaardeBlauw = 0;

        for (i=1; i<1000; i++) { // doe het 1000 keer om de spelers
            // een beetje tijd te geven
            PotMeterStand = analogRead(PotMeter); // lees de stand van de potmeter

            SpeelTijd = map(PotMeterStand,0,1024,60,5); // vertaal het naar een bruikbare
            // waarde voor de speeltijd
            VierCijfers.showNumberDec (SpeelTijd, false,2, 1); // laat de speeltijd zien

            // ==== PLAATS 4 begin ==== Hieronder komen straks de regels voor de toegift =====
            if (digitalRead(KnopKlein) == LOW) { // Als het kleine knopje ingedrukt is
                // dan gaan we het rood van de LED
                // veranderen
                LEDWaardeBegin = LEDMiddenWaardeRood; // onthoud de beginwaarde van de LED
                PotMeterStandBegin = analogRead(PotMeter); // lees de stand van de potmeter
                while (digitalRead(KnopKlein) == LOW) { // Doe dit zolang de knop is ingedrukt
                    PotMeterStand = analogRead(PotMeter); // lees de nieuwe stand van de
                    // potmeter
                    LEDWaardeNieuw = LEDWaardeBegin
                        + map(PotMeterStand-PotMeterStandBegin,-1024, 1024, 260, -260);
                    // verander de waarde van de LED.
                    // Zie de Bijlage C in de handleiding
                    // voor de uitleg
                    LEDWaardeNieuw = constrain(LEDWaardeNieuw, 0, 255);
                    // Begrens de waarde tot een getal
                    // tussen 0 en 255
                    analogWrite(LEDMiddenRood, LEDWaardeNieuw); // Zet de nieuwe waarde voor de LED

                    VierCijfers.showNumberDec(LEDWaardeNieuw, false,4, 0);
                    // Laat de waarde van de LED zien
                }

                // We hebben het display gebruikt om de waarde van de LED aan te geven
                // Dat moeten we nu weer uitpoetsen, want na het loslaten van de knop gaan we
                // weer verder met het instellen van de speeltijd.
            }
        }
    }
}

```

```

VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

LEDMiddenWaardeRood = LEDWaardeNieuw;          // Onthoud de nieuwe waarde van de LED
                                                // voor een volgende ronde

i = i - 200;                                    // Geef een beetje extra tijd
                                                // door de teller terug te zetten
}

if (digitalRead(KnopRechts) == LOW) {           // Als de rechter knop ingedrukt is
                                                // dan gaan we het blauw van de LED
                                                // veranderen
LEDWaardeBegin = LEDMiddenWaardeBlauw;          // onthoud de beginwaarde van de LED
PotMeterStandBegin = analogRead(PotMeter);      // lees de stand van de potmeter
while (digitalRead(KnopRechts) == LOW) {        // Doe dit zolang de knop is ingedrukt
    PotMeterStand = analogRead(PotMeter);        // lees de nieuwe stand van de
                                                // potmeter

    LEDWaardeNieuw = LEDWaardeBegin
        + map(PotMeterStand-PotMeterStandBegin,-1024, 1024, 260, -260);
                                                // verander de waarde van de LED.
                                                // Zie de Bijlage C in de handleiding
                                                // voor de uitleg
    LEDWaardeNieuw = constrain(LEDWaardeNieuw, 0, 255);
                                                // Begrens de waarde tot een getal
                                                // tussen 0 en 255
    analogWrite(LEDMiddenBlauw, LEDWaardeNieuw); // Zet de nieuwe waarde voor de LED

    VierCijfers.showNumberDec(LEDWaardeNieuw, false,4, 0);
                                                // Laat de waarde van de LED zien
}

// We hebben het display gebruikt om de waarde van de LED aan te geven
// Dat moeten we nu weer uitpoetsen, want na het loslaten van de knop gaan we
// weer verder met het instellen van de speeltijd.
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

LEDMiddenWaardeBlauw = LEDWaardeNieuw;          // Onthoud de nieuwe waarde van de LED
                                                // voor een volgende ronde

i = i - 200;                                    // Geef een beetje extra tijd
                                                // door de teller terug te zetten
}

if (digitalRead(KnopLinks) == LOW) {           // Als de linker knop ingedrukt is
                                                // dan gaan we het groen van de LED
                                                // veranderen
LEDWaardeBegin = LEDMiddenWaardeGroen;          // onthoud de beginwaarde van de LED
PotMeterStandBegin = analogRead(PotMeter);      // lees de stand van de potmeter
while (digitalRead(KnopLinks) == LOW) {        // Doe dit zolang de knop is ingedrukt
    PotMeterStand = analogRead(PotMeter);        // lees de nieuwe stand van de
                                                // potmeter

    LEDWaardeNieuw = LEDWaardeBegin
        + map(PotMeterStand-PotMeterStandBegin,-1024, 1024, 260, -260);
                                                // verander de waarde van de LED.
                                                // Zie de Bijlage C in de handleiding
                                                // voor de uitleg
    LEDWaardeNieuw = constrain(LEDWaardeNieuw, 0, 255);
                                                // Begrens de waarde tot een getal
                                                // tussen 0 en 255
    analogWrite(LEDMiddenGroen, LEDWaardeNieuw); // Zet de nieuwe waarde voor de LED

    VierCijfers.showNumberDec(LEDWaardeNieuw, false,4, 0);
                                                // Laat de waarde van de LED zien
}

// We hebben het display gebruikt om de waarde van de LED aan te geven
// Dat moeten we nu weer uitpoetsen, want na het loslaten van de knop gaan we
// weer verder met het instellen van de speeltijd.
VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers

LEDMiddenWaardeGroen = LEDWaardeNieuw;          // Onthoud de nieuwe waarde van de LED

```

```

// voor een volgende ronde

    i = i - 200; // Geef een beetje extra tijd
                // door de teller terug te zetten
}

// ==== PLAATS 4 eind ===== Hierboven komen straks de regels voor de toegift =====

}

// ==== PLAATS 3 eind ===== Voor deze regel komt straks het instellen van de speeltijd =

// ==== Voorbereidingen voor het spel =====

StartTijd = millis(); // Onthoud hoe laat het nu is
PuntenLinks = 0;      // Zet de punten van de linker speler op
                        // nul
VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
                                                        // speler in de twee linker cijfers
PuntenRechts = 0;     // Zet de punten van de linker speler op
                        // nul
VierCijfers.showNumberDec(PuntenRechts, false, 2, 2); // Vertoon het aantal punten van de
rechter // speler in de twee rechter cijfers
StatusKnopLinksNu = digitalRead(KnopLinks); // Lees de status van de linker knop
StatusKnopLinksLaatst = StatusKnopLinksNu; // En dat is dan ook de beginwaarde

StatusKnopRechtsNu = digitalRead(KnopRechts); // Lees de status van de linker knop
StatusKnopRechtsLaatst = StatusKnopRechtsNu; // En dat is dan ook de beginwaarde

while ((millis() < StartTijd + SpeelTijd * 1000) && (PuntenLinks < 99) && (PuntenRechts
< 99) ) {
    // Als de huidige tijd (millis()) kleiner
    // is dan de starttijd plus de speeltijd
    // (die we met 1000 vermenigvuldigen om
    // ook milliseconden te krijgen), dan
    // kunnen we doorgaan: de speeltijd is
    // nog niet om.
    // Maar als de speler al 99 punten heeft
    // dan moeten we toch stoppen, want
    // meer dan 99 punten kunnen we niet
    // laten zien.

    // ==== PLAATS 1 begin ===== Hieronder komen straks regels voor de knoppen =====
    StatusKnopLinksNu = digitalRead(KnopLinks);
    if (StatusKnopLinksNu == LOW && StatusKnopLinksLaatst == HIGH) {
        // Als de knop nu LOW is en hij was HIGH
        // in de vorige keer dat we door de lus
        // kwamen, dan is de knop nu net ingedrukt
        PuntenLinks = PuntenLinks + 1; // De linker speler krijgt er een punt bij
        VierCijfers.showNumberDec(PuntenLinks, false, 2, 0); // Vertoon het aantal punten van de linker
                                                                // speler in de twee linker cijfers
    }
    StatusKnopLinksLaatst = StatusKnopLinksNu; // Onthoud de waarde voor de volgende keer
                                                // dat we door de lus komen

    StatusKnopRechtsNu = digitalRead(KnopRechts);
    if (StatusKnopRechtsNu == LOW && StatusKnopRechtsLaatst == HIGH) {
        // Als de knop nu LOW is en hij was HIGH
        // in de vorige keer dat we door de lus
        // kwamen, dan is de knop nu net ingedrukt
        PuntenRechts = PuntenRechts + 1; // De rechter speler krijgt er een punt
bij
        VierCijfers.showNumberDec(PuntenRechts, false, 2, 2); // Vertoon het aantal punten van de
rechter // speler in de twee rechter cijfers
    }
    StatusKnopRechtsLaatst = StatusKnopRechtsNu; // Onthoud de waarde voor de volgende
keer
                                                // dat we door de lus komen

```

```

// ==== PLAATS 1 eind ==== Voor deze regel komen de opdrachten voor de knoppen =====
}

// ==== PLAATS 2 begin ==== Hieronder komen straks de regels voor de winnaar =====
// Deze ronde is afgelopen. Laat nu de punten van de winnaar, of van beide bij gelijk
// spel knipperen. Dat doen we door eerst de cijfers van de winnaar (of beiden) weg te
// halen en dan beide weer te laten zien.
while (digitalRead(KnopKlein) == HIGH) {
    // Als de kleine knop wordt
    // ingedrukt dan stoppen we met
    // te laten zien wie heeft
    // gewonnen.

    if (PuntenLinks >= PuntenRechts) {
        // Als Links heeft gewonnen of als
        // het gelijk is ...
        VierCijfers.setSegments(TweeCijfersLeeg,2,0); // Wis de linker 2 cijfers
    }

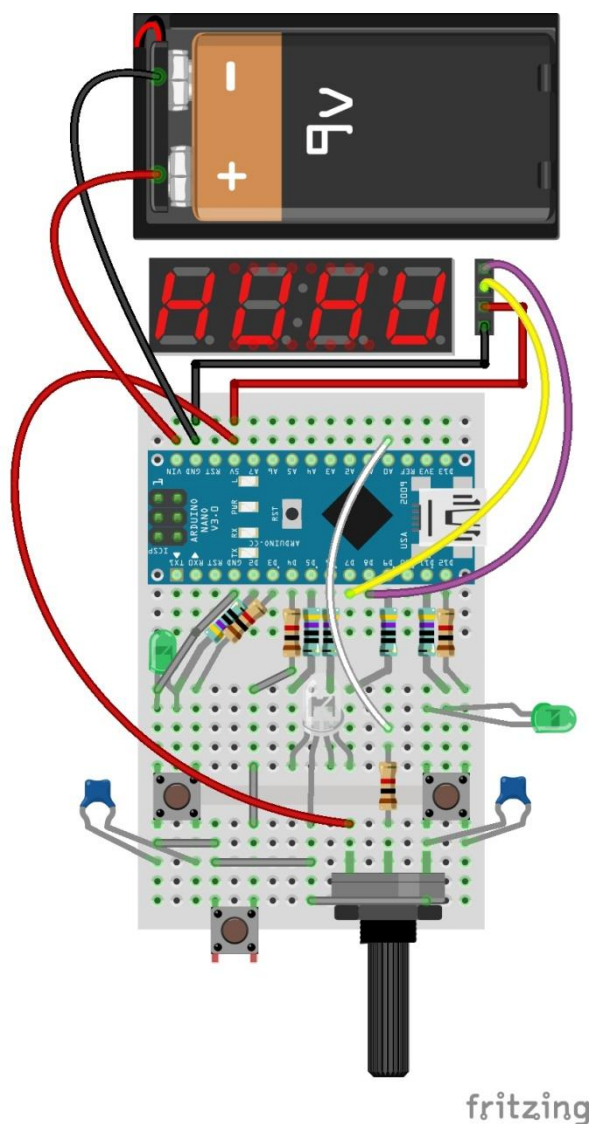
    if (PuntenRechts >= PuntenLinks) {
        VierCijfers.setSegments(TweeCijfersLeeg,2,2); // Wis de rechter 2 cijfers
    }

    delay (100);
    VierCijfers.showNumberDec(PuntenLinks, false, 2, 0);
    VierCijfers.showNumberDec(PuntenRechts, false, 2, 2);
    delay (100);
}

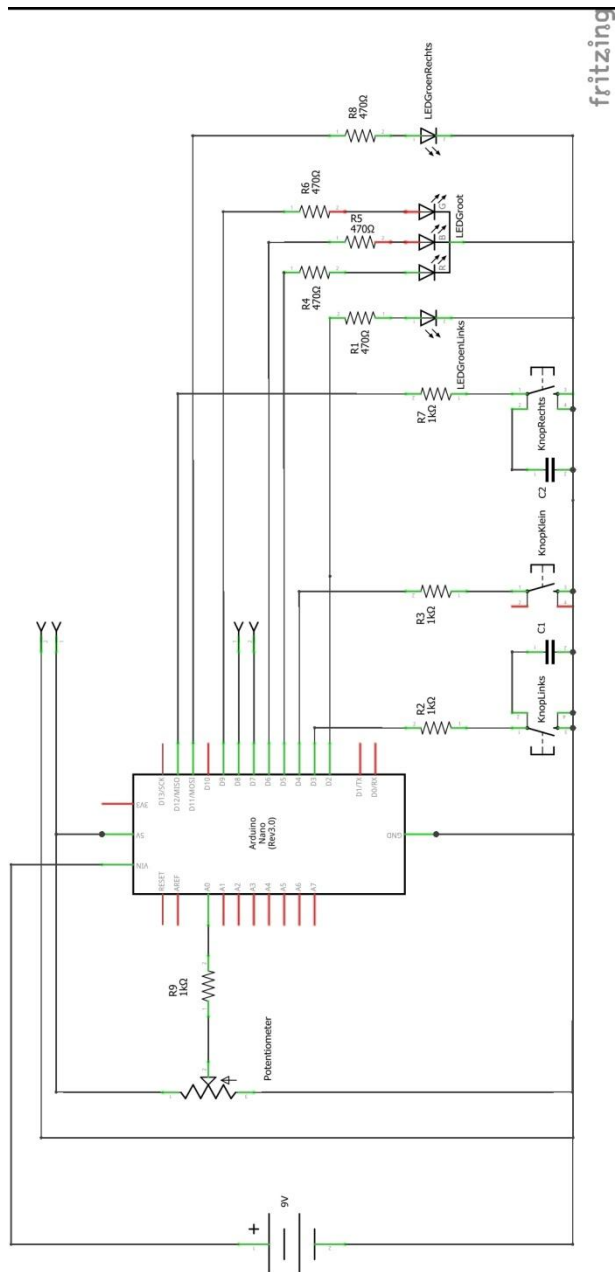
// ==== PLAATS 2 eind ==== Hierboven komen straks de regels voor de winnaar =====
}
}
}

```

De leerlingen vragen mogelijk naar de schakeling. Daarom hieronder een uitleg van de schakeling.

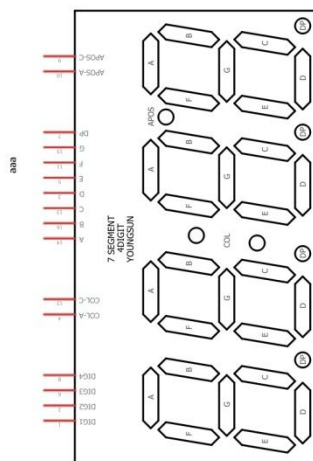


Figuur 1 - Het bedradingsschema



fritzing

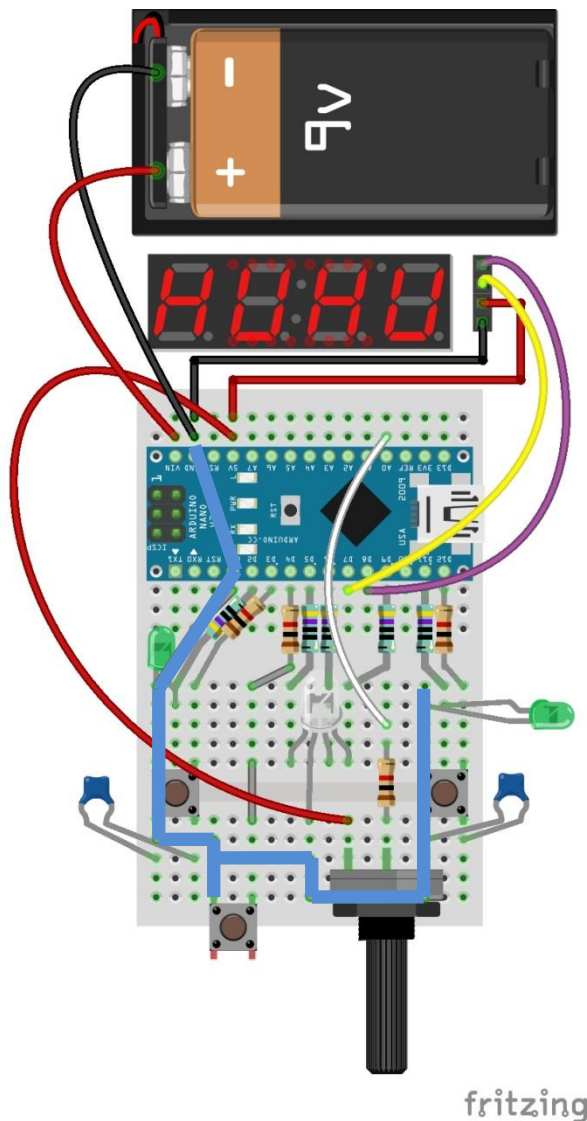
Figuur 2. Het logische schema



Voor algemene vragen over hoe Arduino werkt, zie het (nog te schrijven) document *Elektronica rondom Arduino*. Bijvoorbeeld vragen als:

- Hoe werkt een breadboard?
- Waarom een serieweerstand met de LED?
- Waarom een serieweerstand met de schakelaars?
- Hoe werkt de waarneming van een schakelaar?
- Hoe bereken je de waarde van de weerstanden?
- Hoe werkt de anti-dender condensator?
- Hoe bereken je hoe groot die moet zijn?
- Hoe werkt de analoge input van de potentiometer
- Waarom een weerstand in serie met het middencontact van de potentiometer?

Specifiek voor de ze les is de vraag: Hoe loopt het GND signaal? Dat loopt namelijk door de schakelaars heen en dat is niet zo voor de hand liggend. Zie de dikke blauwe lijnen.



WAT KOST EEN PLANKJE?

Het kan zijn dat de leerlingen vragen wat het plankje met de schakeling dat ze meekrijgen kost. Uiteraard hoeven ze er niet voor te betalen. Maar ze mogen wel weten wat het ongeveer kost, om te zorgen dat ze begrijpen dat het wel waardevol is, maar dat het ook geen fortuin is wat ze meekrijgen.

----- Hier een tabel met de kosten van de onderdelen.