

Trapspanningsgenerator

Studenten: Daan Dekoning Krekels, Jesse Denaux

Berekeningen

$$V_{out} = -(R_f/R_1 \cdot V_{in1} + R_f/R_2 \cdot V_{in2} + \dots R_f/R_n \cdot V_{inn})$$

$$R_f = 10\text{k}\Omega$$

$$R_f/R_1 = 0.1$$

$$R_1 = 10\text{k}/0.1 = 100\text{k}\Omega$$

$$R_f/R_2 = 0.2$$

$$R_2 = 50\text{k}\Omega$$

$$R_f/R_3 = 0.4$$

$$R_3 = 25\text{k}\Omega$$

$$R_f/R_4 = 0.8$$

$$R_4 = 12.5\text{k}\Omega$$

$$0000 = 0\text{V}$$

$$0001 = 0.5\text{V}$$

$$0010 = 1\text{V}$$

$$0011 = 1.5\text{V}$$

$$0100 = 2\text{V}$$

$$0101 = 2.5\text{V}$$

$$0110 = 3\text{V}$$

$$0111 = 3.5\text{V}$$

$$1000 = 4\text{V}$$

$$1001 = 4.5\text{V}$$

$$1010 = 5\text{V}$$

$$1011 = 5.5\text{V}$$

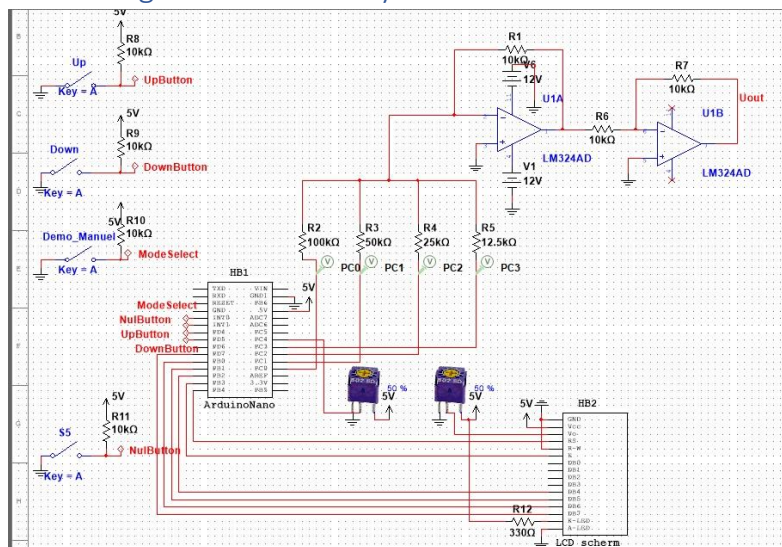
$$1100 = 6\text{V}$$

$$1101 = 6.5\text{V}$$

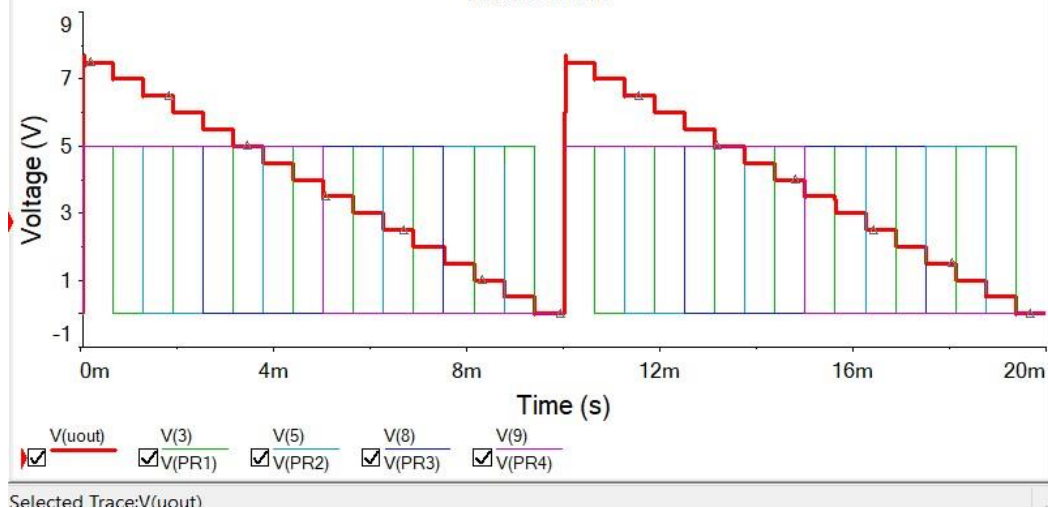
$$1110 = 7\text{V}$$

$$1111 = 7.5\text{V}$$

Schakeling + transient analyse



trapversterkingsgenerator Transient



Handleiding

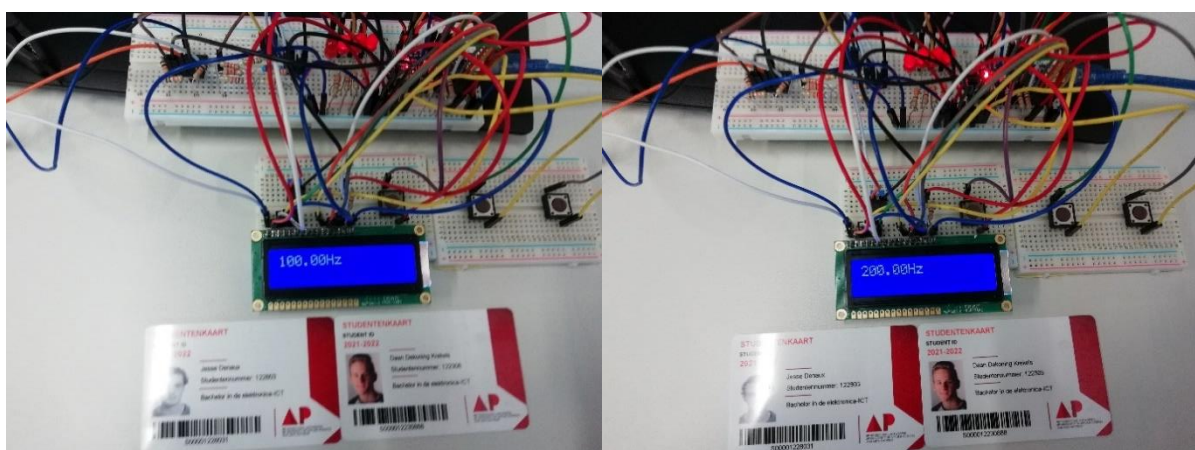
Onderstaande link is een filmpje waar alles visueel zichtbaar is:

<https://ap.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=8960343f-da4f-4244-b444-ae410116d925>

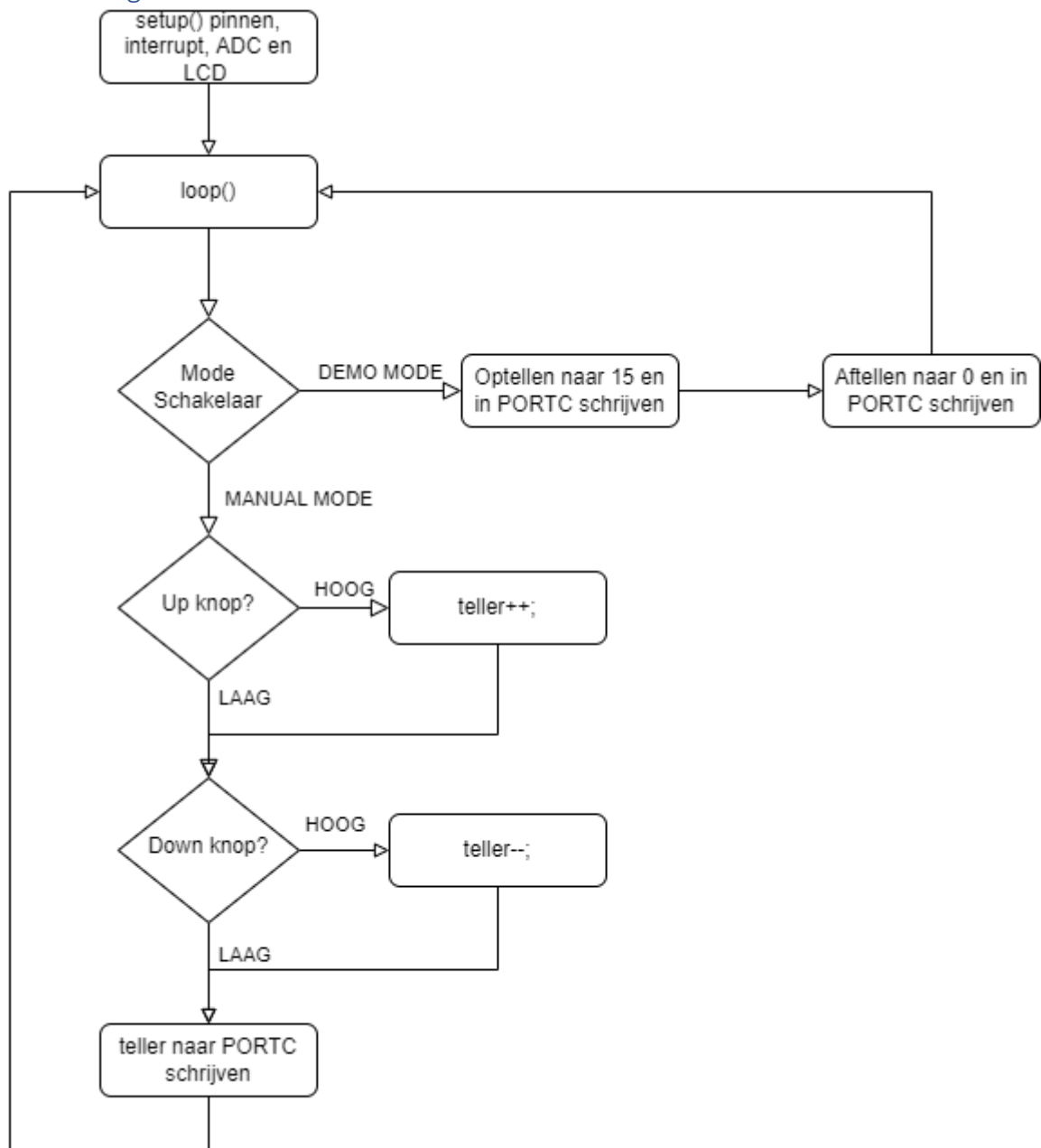
Wanneer de schakelaar van de demo/manuele toestand aanstaat dan staat de schakeling in demo mode. Dit wil zeggen dat deze van 0V naar 7.5V gaat en terug naar 0V in stappen van 0.5V. Wordt de schakelaar uitgezet dan komt men in manuele mode. Hierbij is er 0V aan de uitgang. De gebruiker kan dan aan de hand van 2 knoppen de uitgangsspanning met 0.5V verhogen tot 7.5V of verlagen tot 0V. Er is ook nog een 3e knop waarbij de uitgangsspanning naar nul wordt herleid.



Het display toont de frequentie van de schakeling. Deze frequentie kan ook nog worden aangepast aan de hand van een potentiometer tussen 100Hz en 200Hz.



Stroomdiagram Code



Setup

```
1 void setup()
2 {
3   Serial.begin(9600);
4   Serial.println("Compiled: " __FILE__
5                 ", " __DATE__ ", " __TIME__ ", " __VERSION__);
6
7   /*
8    * Pin instellingen
9    */
10  DDRC = 0b00001111; // Uitgang naar somversterker en POT op A4
11  DDRB = 0b00011111; // LCD pinnen
12  DDRD = 0b10000000; // LCD en buttons
13
14  /*
15   * Interrupt pin om naar 0 te zetten
16   */
17  EIMSK |= (1 << INT1);
18  // Neergaande flank
19  EICRA |= (1 << ISC11);
20  EICRA &= ~(1 << ISC10);
21
22  /*
23   * ADC op A4 om gewenste frequentie in te lezen
24   */
25
26  ADCSRA |= ((1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0)); //Prescaler at 128 so we have an 125Khz clock source
27
28  ADMUX |= (1 << REFS0);
29  ADMUX &= ~(1 << REFS1); //Avcc(+5v) as voltage reference
30
31  ADMUX &= ~(1 << MUX3); // 0100 ADC4
32  ADMUX |= (1 << MUX2);
33  ADMUX &= ~(1 << MUX1);
34  ADMUX &= ~(1 << MUX0);
35
36  ADCSRB &= ~(1 << ADTS2) | (1 << ADTS1) | (1 << ADTS0)); //ADC in free-running mode
37  ADCSRA |= (1 << ADIFSC); //Signal source, in this case is the free-running
38  ADCSRA |= (1 << ADIFRNG); //Power up the ADC
39  ADCSRA |= (1 << ADSC);
40
41  /*
42   * LCD initialiseren
43   */
44  lcd.begin(16, 2);
45 }
```

Loop demo mode

```
1 // Kies mode afhankelijk van de MODE_SELECT schakelaar
2 if ((PIND & MODE_SELECT) == MODE_SELECT)
3 {
4     // DEMO MODE
5     // Optellen en weer aftellen
6     Serial.println("DEMO MODE");
7     for (byte i = 0; i <= 15; i++)
8     {
9         PORTC = i;
10        // Serial.println(PORTC,BIN);
11        Serial.println(PORTC);
12        delay(snelheid()); // Bereken delay snelheid voor juiste frequentie
13    }
14    for (byte i = 14; i > 0; i--)
15    {
16        PORTC = i;
17        // Serial.println(PORTC,BIN);
18        Serial.println(PORTC);
19        delay(snelheid());
20    }
21 }
```

```
1 int snelheid()
2 {
3     // Start me ADC te lezen
4     ADCSRA |= (1 << ADSC);
5     while (!(ADCSRA & (1 << ADIF)))
6     {
7     } // hier geen instructies, gewoon wachten.
8     byte tLaag = ADCL;
9     byte tHoog = ADCH; // Combineren van ADC bytes naar in en die meten mappen naar een waarde tussen 5 of 10ms
10    byte tijd = map((tHoog << 8) + tLaag, 0, 1023, 5, 10);
11    // De frequentie is 1/t
12    float frec = 1.0 / (tijd / 1000.0); // tijd eerst ms → seconden
13    lcd.clear(); // Frequentie weergeven op LCD
14    lcd.setCursor(0, 0);
15    lcd.print(frec);
16    lcd.print("Hz");
17    Serial.print(frec);
18    return tijd;
19 }
```

Loop manual mode

```
1  else
2  {
3      // MANUAL MODE
4      if ((PIND & UP_BTN) == UP_BTN)
5      {
6          up = true;
7          delay(100); // Voorkom te snel achter elkaar drukken
8      }
9      if (up)
10     {
11         if (teller < 15)
12         {
13             teller++;
14         }
15         up = false;
16     }
17     if ((PIND & DOWN_BTN) == DOWN_BTN)
18     {
19         down = true;
20         delay(100); // Voorkom te snel achter elkaar drukken
21     }
22     if (down)
23     {
24         if (teller > 0)
25         {
26             teller--;
27         }
28         down = false;
29     }
30     Serial.println(teller);
31     PORTC = teller;
32 }
33
```

Reset knop

```
1  ISR(INT1_vect)
2  {
3      // Interrupt om de teller te resetten
4      teller = 0;
5      PORTC = teller;
6  }
```

Volledige code

```
/* Trapversterkingsgenerator
 * Jesse Denaux en Daan Dekoning Krekels
 * 17/02/2022
 */

#include <LiquidCrystal.h>
#define MODE_SELECT 0b000000100
#define ZERO_BTN 0b000001000 // INT1
#define UP_BTN 0b000010000 // PD4
#define DOWN_BTN 0b000100000 // PD5

byte teller = 0;
bool up = false;
bool down = false;
// LCD bibliotheek initialiseren
const int rs = 12, en = 11, d4 = 10, d5 = 9, d6 = 8, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()
{
    Serial.begin(9600);
    Serial.println("Compiled: " __FILE__
                  ", " __DATE__ ", " __TIME__ ", " __VERSION__);

    /*
     * Pin instellingen
     */
    DDRC = 0b00001111; // Uitgang naar somversterker en POT op A4
    DDRB = 0b00011111; // LCD pinnen
    DDRD = 0b10000000; // LCD en buttons

    /*
     * Interrupt pin om naar 0 te zetten
     */
    EIMSK |= (1 << INT1);
```



```

// Neergaande flank
EICRA |= (1 << ISC11);
EICRA &= ~(1 << ISC10);

/*
 * ADC op A4 om gewenste frequentie in te lezen
 */

ADCSRA |= ((1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0)); //Prescaler
at 128 so we have an 125Khz clock source

ADMUX |= (1 << REFS0);
ADMUX &= ~(1 << REFS1); //Avcc(+5v) as voltage reference

ADMUX &= ~(1 << MUX3); // 0100 ADC4
ADMUX |= (1 << MUX2);
ADMUX &= ~(1 << MUX1);
ADMUX &= ~(1 << MUX0);

ADCSRB &= ~((1 << ADTS2) | (1 << ADTS1) | (1 << ADTS0)); //ADC in
free-running mode
ADCSRA |= (1 << ADATE); //Signal
source, in this case is the free-running
ADCSRA |= (1 << ADEN); //Power up
the ADC
ADCSRA |= (1 << ADSC);

/*
 * LCD initialiseren
 */
lcd.begin(16, 2);
}

void loop()
{
// Kies mode afhankelijk van de MODE_SELECT schakelaar
if ((PIND & MODE_SELECT) == MODE_SELECT)
{
// DEMO MODE
// Optellen en weer aftellen
Serial.println("DEMO MODE");
for (byte i = 0; i <= 15; i++)
{
PORTC = i;
// Serial.println(PORTC,BIN);
Serial.println(PORTC);
delay(snelheid()); // Bereken delay snelheid voor juiste
frequentie
}
for (byte i = 14; i > 0; i--)
{
PORTC = i;
// Serial.println(PORTC,BIN);
Serial.println(PORTC);
delay(snelheid());
}
}
else

```

```

{
    // MANUAL MODE
    if ((PIND & UP_BTN) == UP_BTN)
    {
        up = true;
        delay(100); // Voorkom te snel achter elkaar drukken
    }
    if (up)
    {
        if (teller < 15)
        {
            teller++;
        }
        up = false;
    }
    if ((PIND & DOWN_BTN) == DOWN_BTN)
    {
        down = true;
        delay(100); // Voorkom te snel achter elkaar drukken
    }
    if (down)
    {
        if (teller > 0)
        {
            teller--;
        }
        down = false;
    }
    Serial.println(teller);
    PORTC = teller;
}
}

ISR(INT1_vect)
{
    // Interrupt om de teller te resetten
    teller = 0;
    PORTC = teller;
}

int snelheid()
{
    // Start me ADC te lezen
    ADCSRA |= (1 << ADSC);
    while (!(ADCSRA & (1 << ADIF)))
    {
    } // hier geen instructies, gewoon wachten.
    byte tLaag = ADCL;
    byte tHoog = ADCH; // Combineren van ADC bytes naar in en die meten
    mappen naar een waarde tussen 5 of 10ms
    byte tijd = map((tHoog << 8) + tLaag, 0, 1023, 5, 10);
    // De frequentie is 1/t
    float frec = 1.0 / (tijd / 1000.0); // tijd eerst ms -> seconden
    lcd.clear(); // Frequentie weergeven op LCD
    lcd.setCursor(0, 0);
    lcd.print(frec);
    lcd.print("Hz");
    Serial.print(frec);
}

```

```
    return tijd;  
}
```