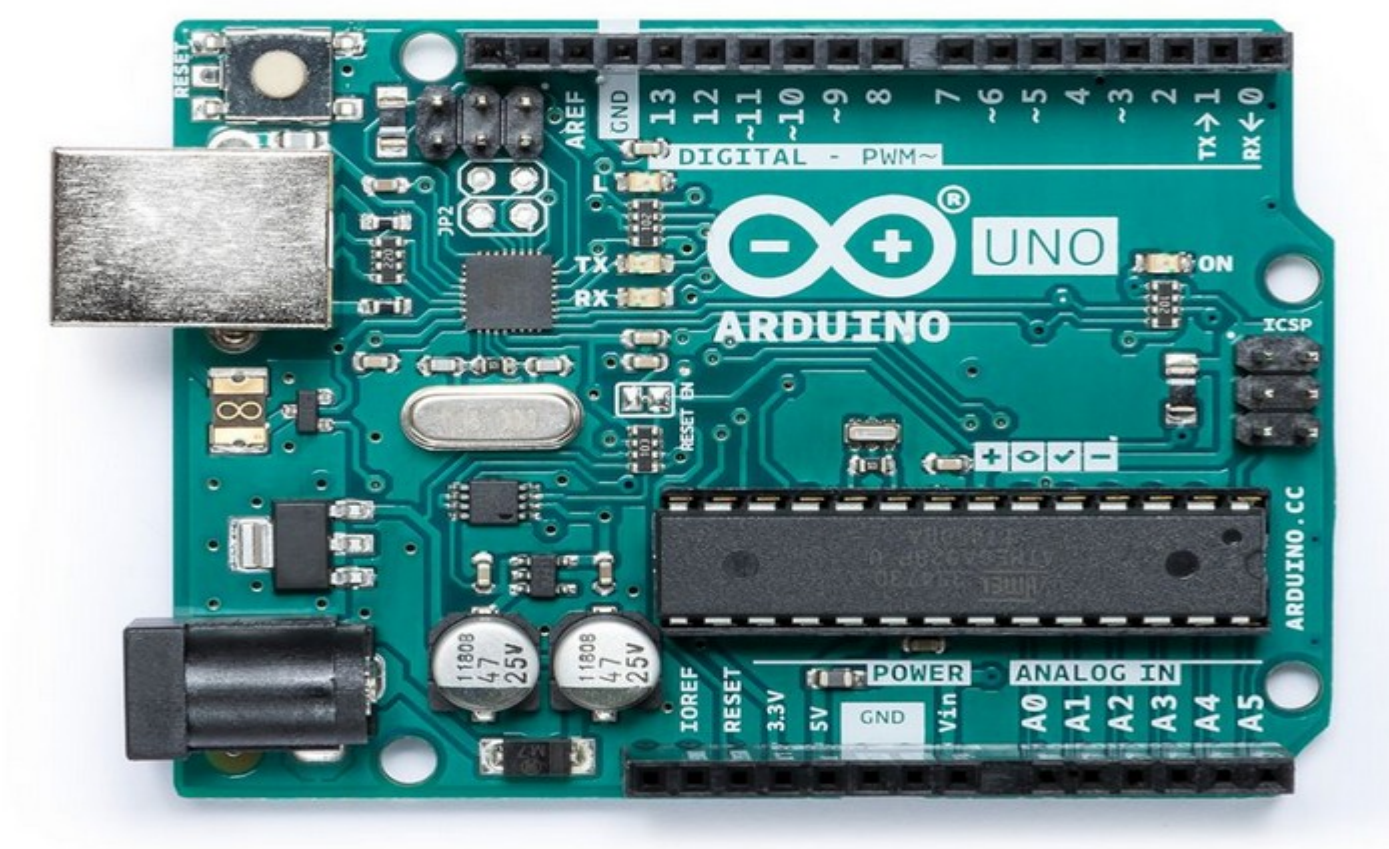
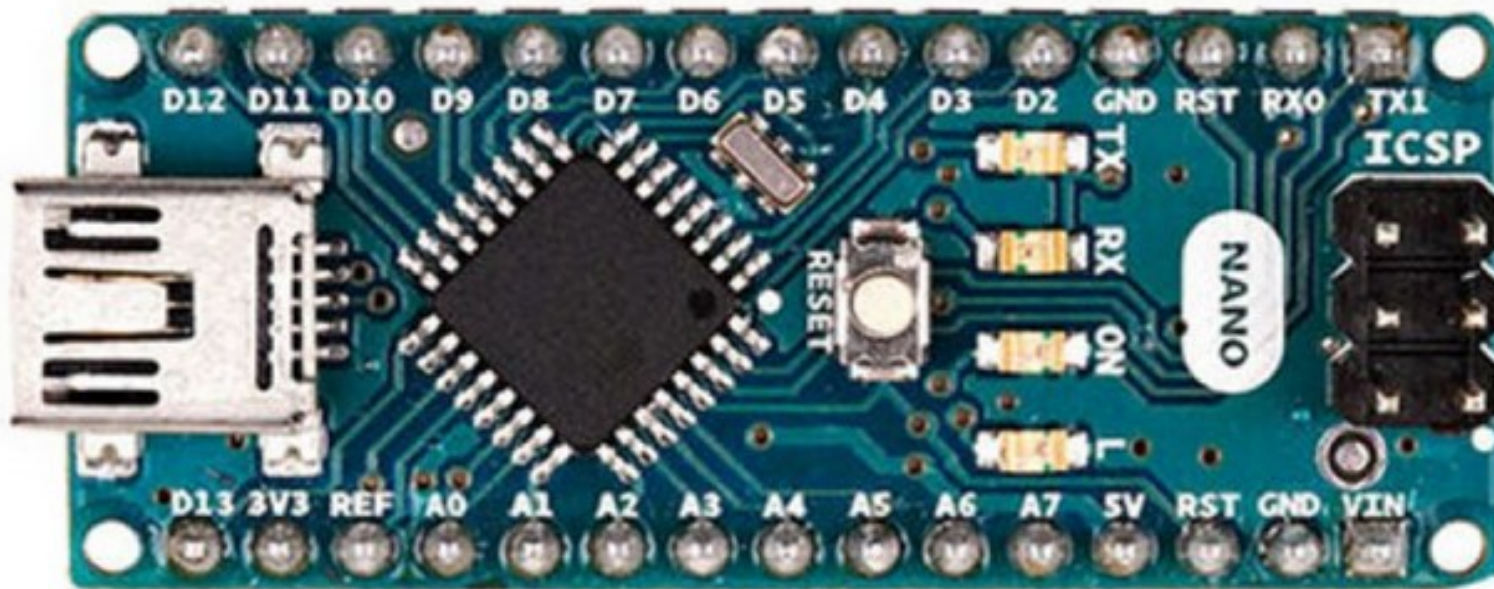


Arduino introductie.



Arduino UNO

Arduino introductie.



Arduino NANO

Arduino introductie.

De Arduino boards zijn heel geschikt voor electronica projecten die maar één taak hoeven te verrichten. Voorbeelden:

- Licht met bewegingsensor.
- Camera systeem met bewegingsensor.
- Sensor gestuurd beregeningsysteem
- Robot wagentje
- Weerstation met sensoren voor temperatuur, luchtvochtigheid, windsnelheid en windrichting.
- Tijd klok met intelligente functies.

Arduino introductie.

De populaire Arduino UNO en NANO gebruiken beide de Atmel ATmega328 microcontroller chip.

De microcontroller beschikt over 32 kB flash geheugen voor het programma en 2 kB sram voor variabelen, ruim voldoende voor allerlei automatisering taken.

Ook is er nog een e-prom deel beschikbaar.

Een klein deel van het geheugen wordt in beslag genomen door de bootloader, een programma waarmee je je gecompileerde programma in de arduino laad.

Arduino introductie.

In de loop van de tijd zijn er veel verschillende boards ontwikkeld. De meeste boards zijn gebaseerd op de Atmel ATmega328P of ATmega168P microcontroller. Ook zijn er veel Chinese klonen op de markt voor heel lage prijzen.

Er is een speciale Arduino IDE beschikbaar om programma's te ontwikkelen, compileren en te uploaden naar het Arduino board.

Daarnaast bieden verschillende leveranciers een op USB gebaseerd programmeer station aan (Tiny USB programmer).

Arduino introductie.

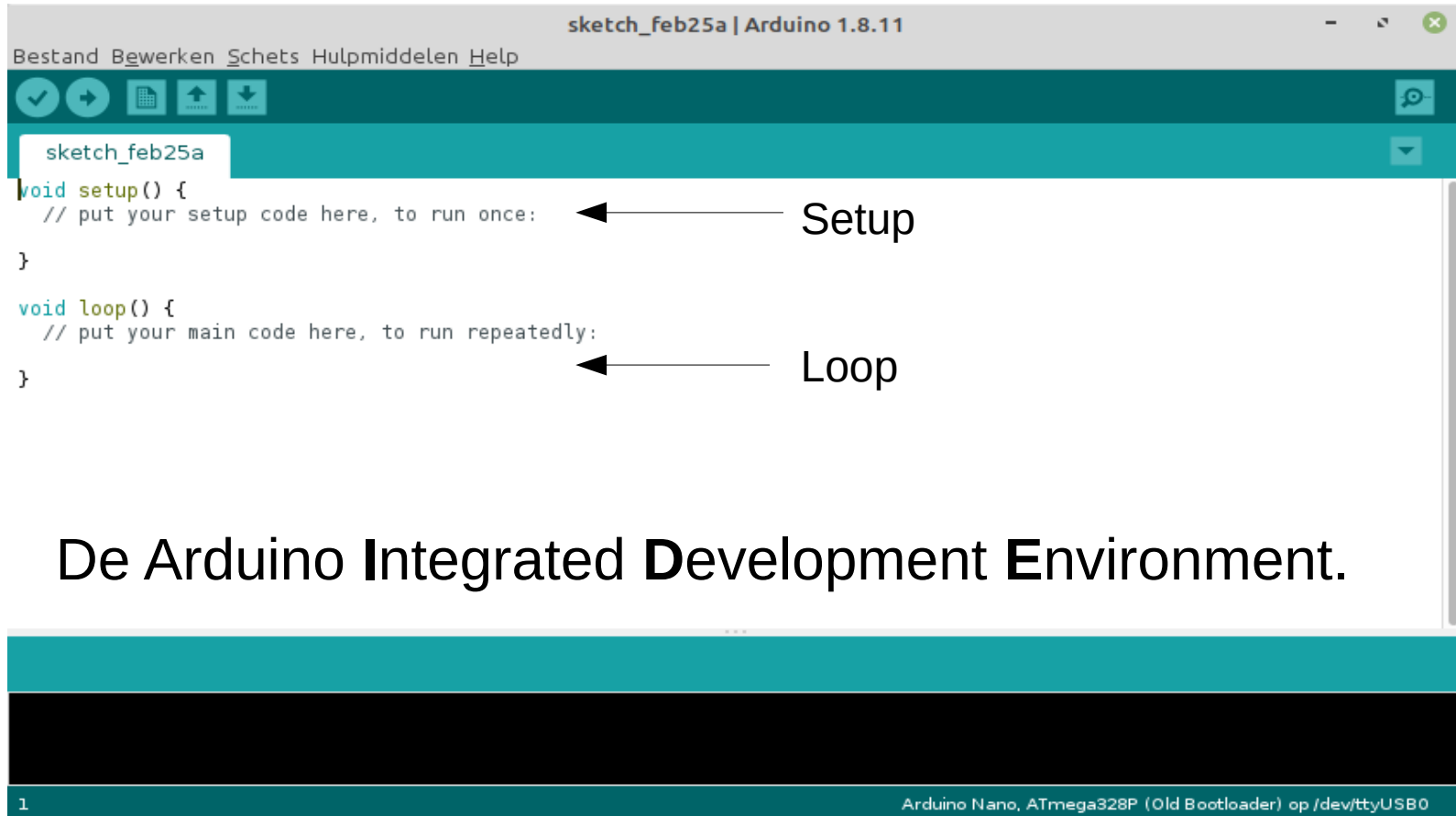
Een Arduino wordt geprogrammeerd in een voor de Arduino aangepaste versie van de programmeertaal C / C++. De Arduino IDE maakt het mogelijk je programma te schrijven, compileren en te uploaden naar de Arduino.

Een Arduino programma bestaat in principe uit drie delen:

- Bibliotheek en variabelen declaraties,
- Setup
- Loop.

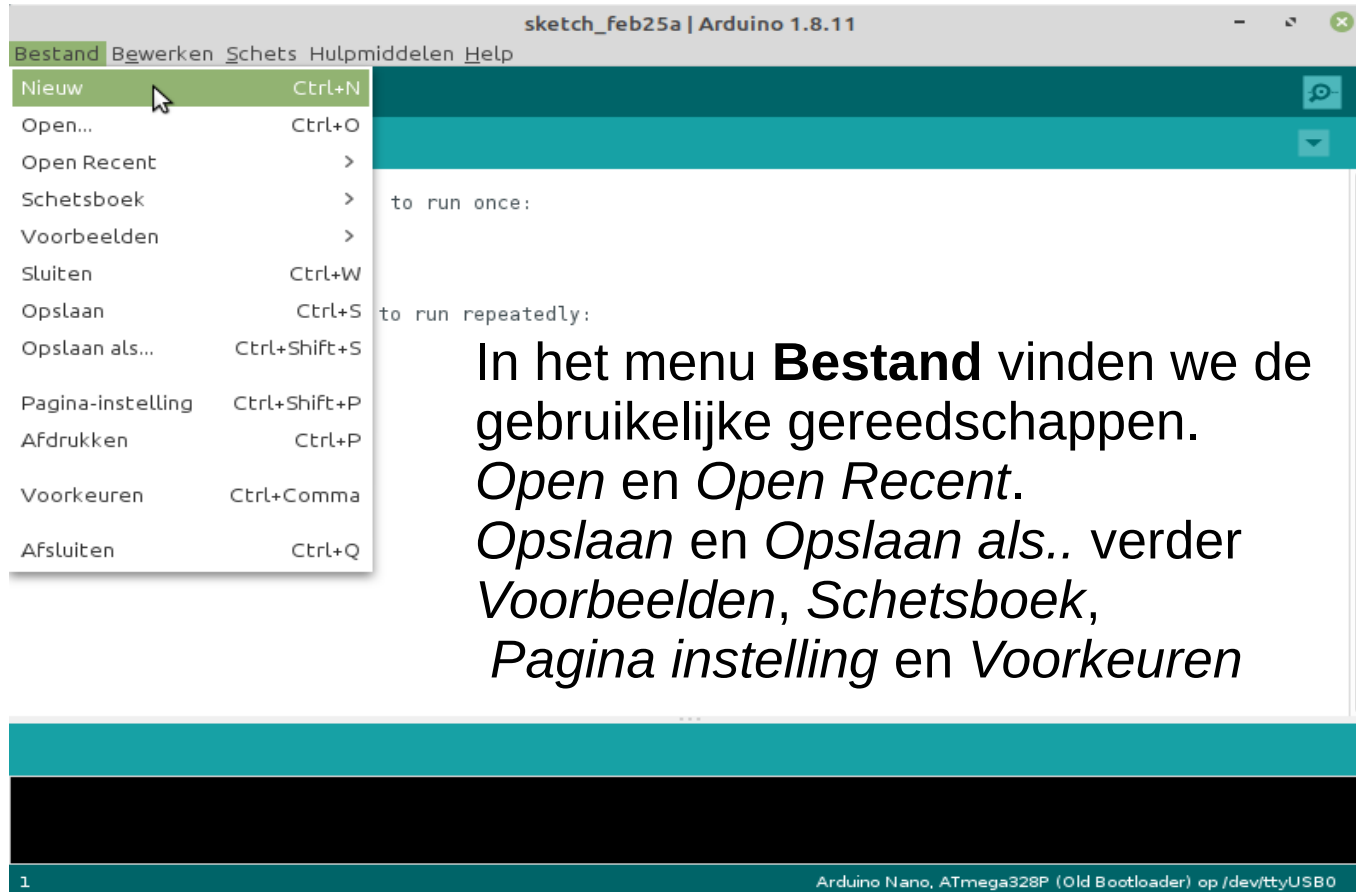
Het declaratie en setup deel wordt éénmaal doorlopen bij de start, het daarop volgende loop deel wordt steeds herhaald.

Arduino introductie.

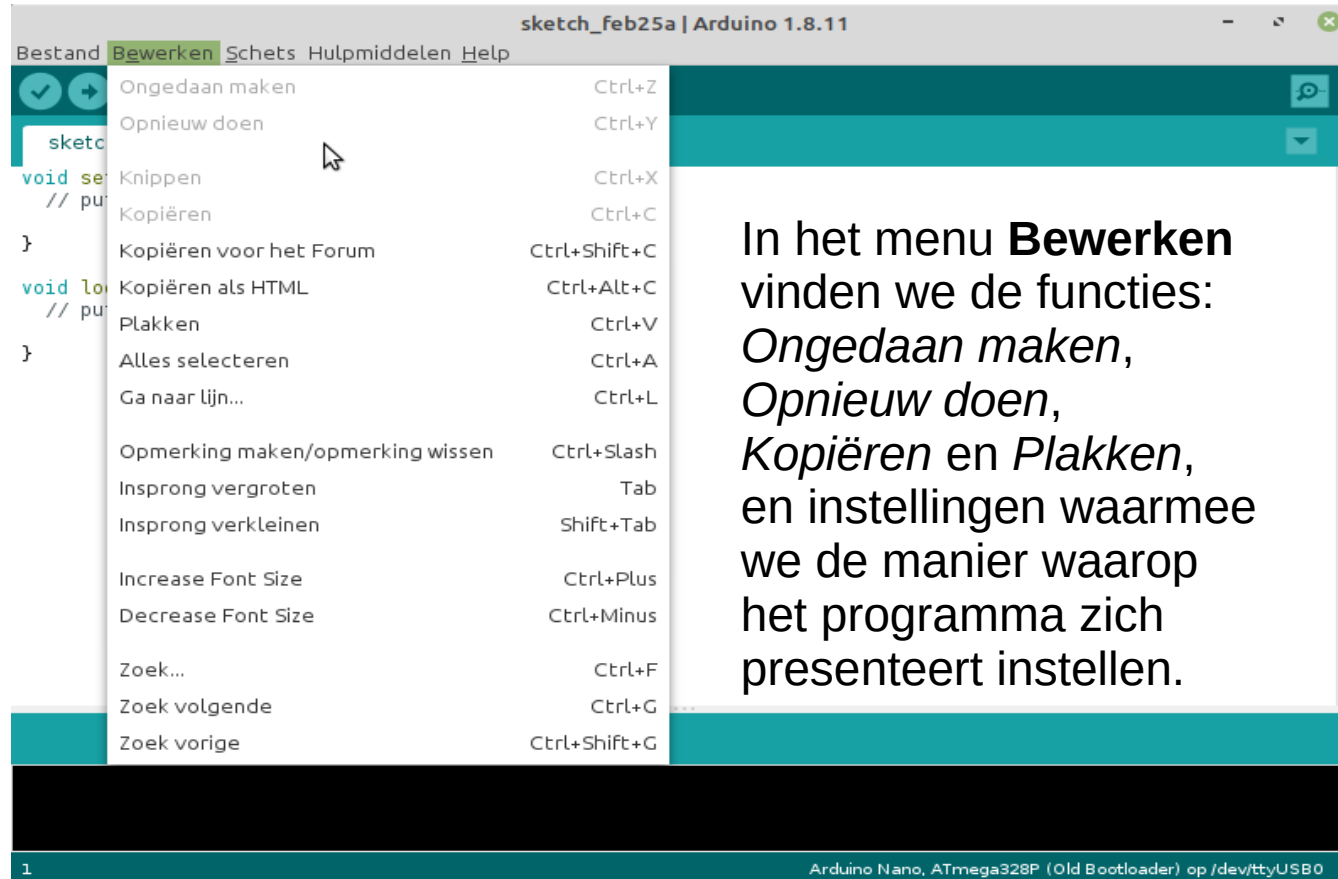


De Arduino **I**ntegrated **D**evelopment **E**nvironment.

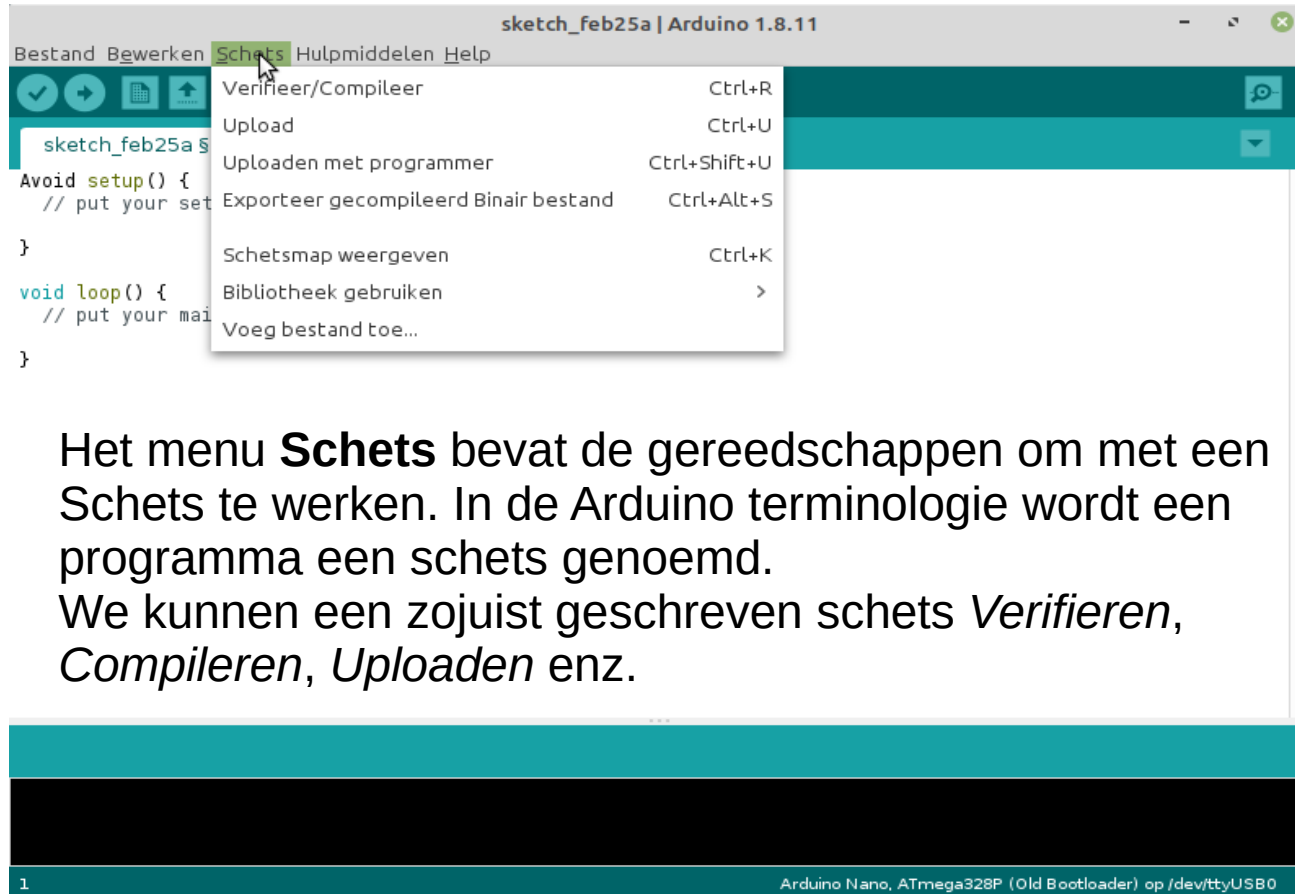
Arduino introductie.



Arduino introductie.



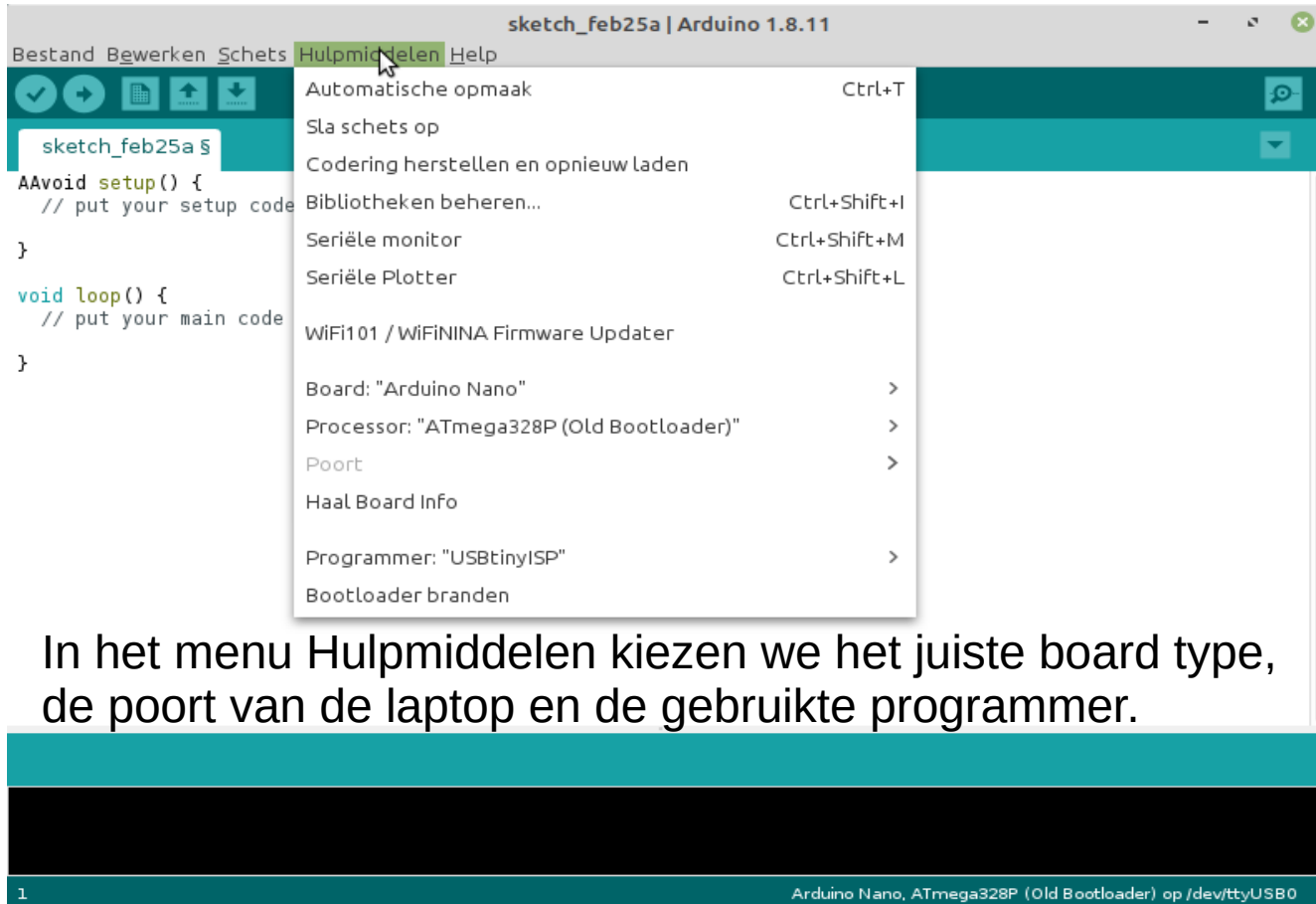
Arduino introductie.



Het menu **Schets** bevat de gereedschappen om met een Schets te werken. In de Arduino terminologie wordt een programma een schets genoemd.

We kunnen een zojuist geschreven schets *Verifiëren*, *Compileren*, *Uploaden* enz.

Arduino introductie.

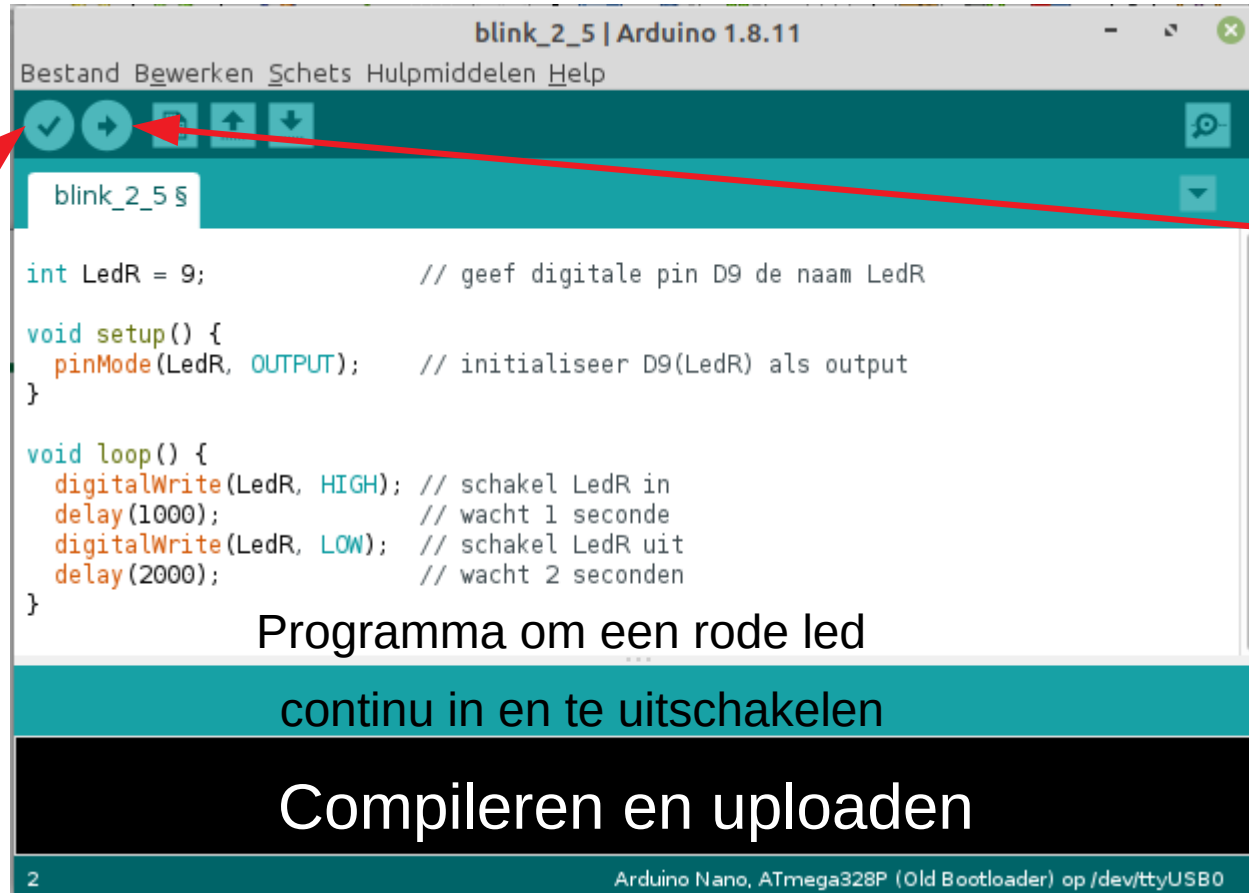


In het menu Hulpmiddelen kiezen we het juiste board type, de poort van de laptop en de gebruikte programmer.

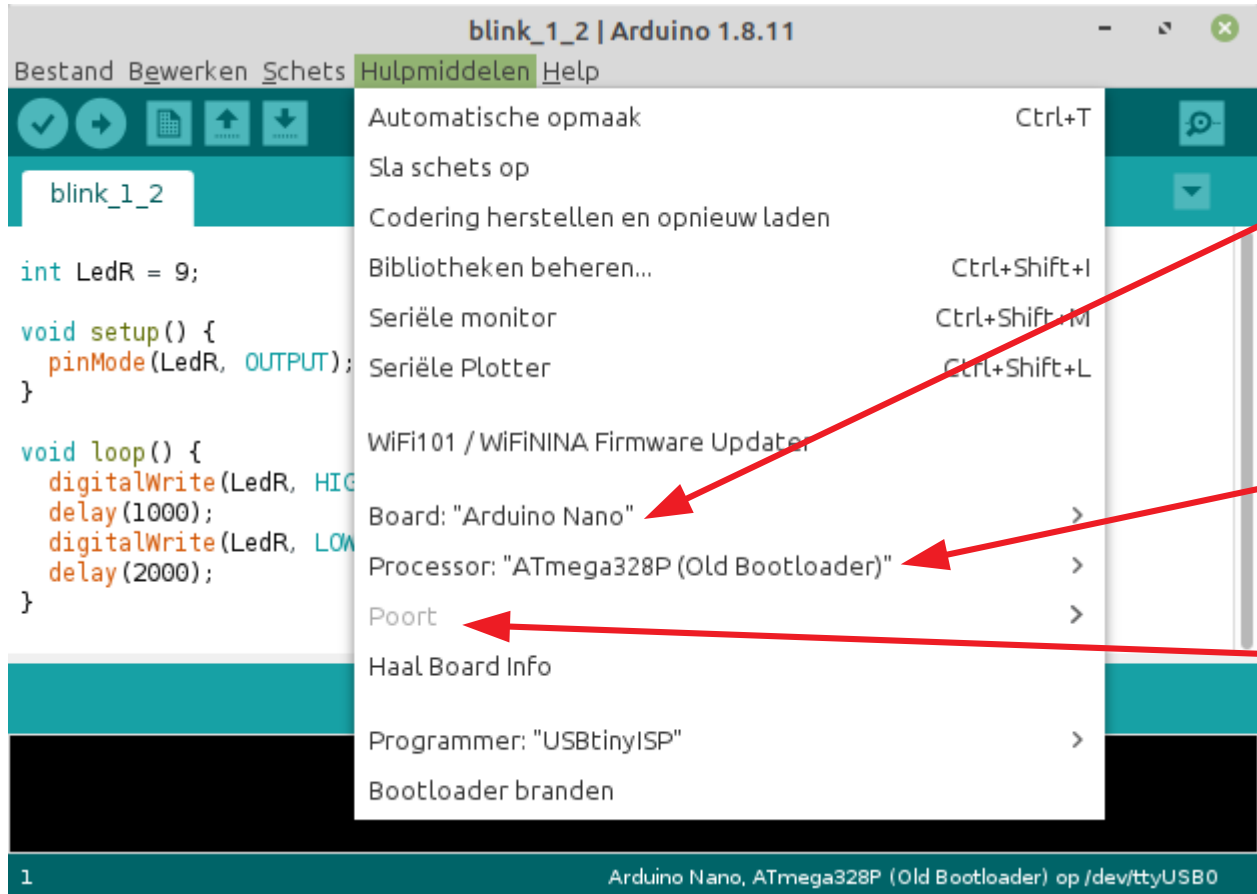
Arduino introductie.

Compileren

Uploaden



Arduino introductie.



Communicatie probleem bij het uploaden?

Staat het juiste board ingesteld?

Is de correcte bootloader gekozen?

Ziet de Arduino IDE jouw Arduino?

Arduino introductie.

Communicatie problemen bij Linux.

Als je Ubuntu of Linux Mint gebruikt om je Arduino te programmeren kun je tegen lastige communicatie problemen aanlopen. Dat heeft o.a. te maken met het stringente bevoegdheden beleid in Linux.

Een systematische aanpak van dit probleem wordt beschreven in een artikel van Majenko Technologies. In dit artikel wordt je stap voor stap door de instellingen geleid.

<https://hackingmajenkoblog.wordpress.com/2016/08/24/diagnosing-arduino-problems-in-linux/>

Arduino introductie.

Sommige Chinese klonen van de Arduino Nano zijn niet voorzien van een bootloader, en je kunt dan ook geen programma rechtstreeks met de IDE uploaden.

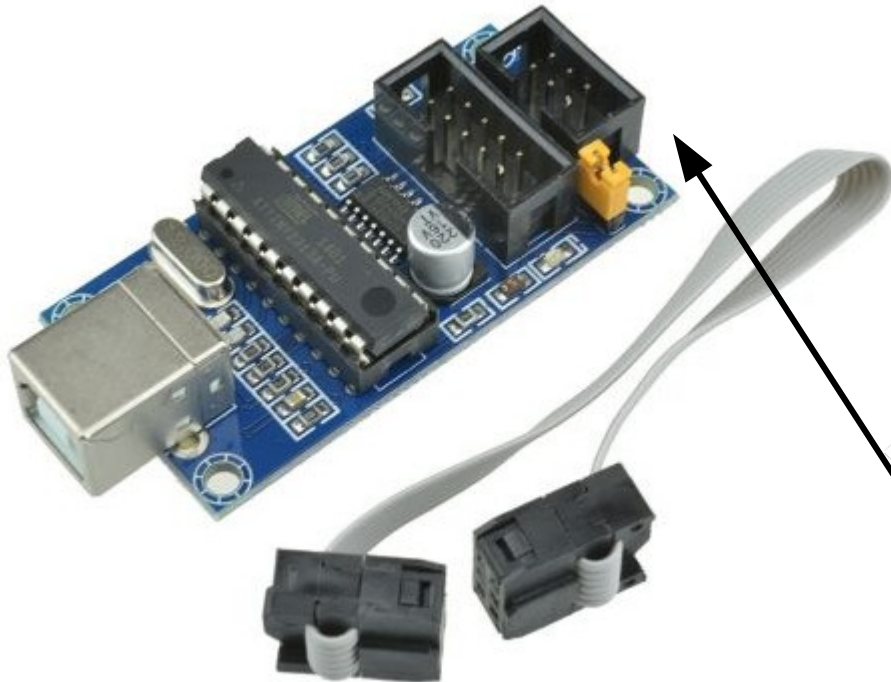
Er zijn verschillende manieren om een bootloader in een dergelijke kloon te "branden". Op internet circuleren veel filmpjes hoe je dat doet.

Je kunt ook van de ISP (In System Programming) aansluiting gebruik maken om je programma te uploaden.

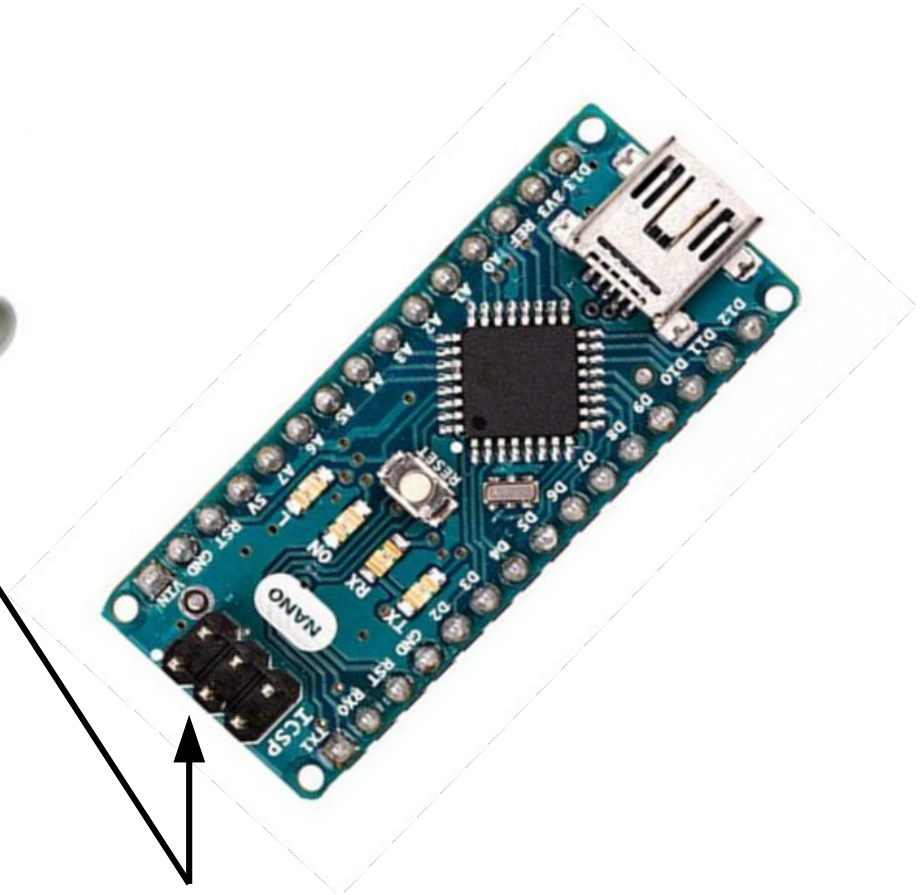
Veel gebruikt is de Tiny USB AVR programmer.

Arduino introductie.

Tiny USB AVR programmer



ISP (In System Programming)

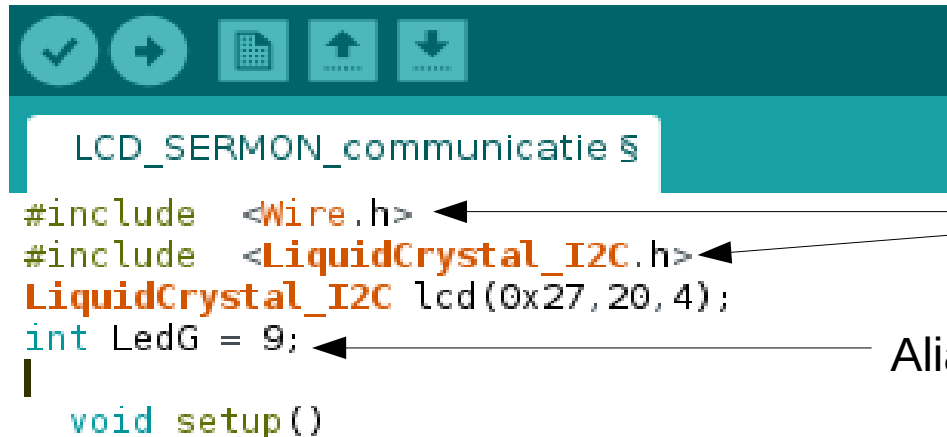


ISP connector

Arduino introductie.

Programma structuur.

- Bibliotheken worden voor de Setup van het programma gedeclareerd.
- Aliassen voor pin codes worden voor Setup toegekend.
- Variabelen (globaal) kunnen voor Setup worden gedeclareerd.



```
LCD_SERMON_communicatie 5
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
int LedG = 9;
void setup()
```

Bibliotheken

Alias LedG voor pin D9

Arduino introductie.

```
blink_1_2 | Arduino 1.8.11
Bestand Bewerken Schets Hulpmiddelen Help

blink_1_2

int LedR = 9;           // geef digitale pin D9 de naam LedR
int sense = 8;          // geef digitale pin D8 de naam sense
int senseniv;           // senseniv variabele voor momentele waarde sense
void setup() {
  pinMode(LedR, OUTPUT); // initialiseer D9(LedR) als output
  pinMode(sense, INPUT);
}

void loop() {
  senseniv = digitalRead(sense);
  if (senseniv == HIGH){
    digitalWrite(LedR, HIGH); // schakel LedR in
    delay(1000);              // wacht 1 seconde
    digitalWrite(LedR, LOW);  // schakel LedR uit
    delay(2000);              // wacht 2 seconden
  }
}

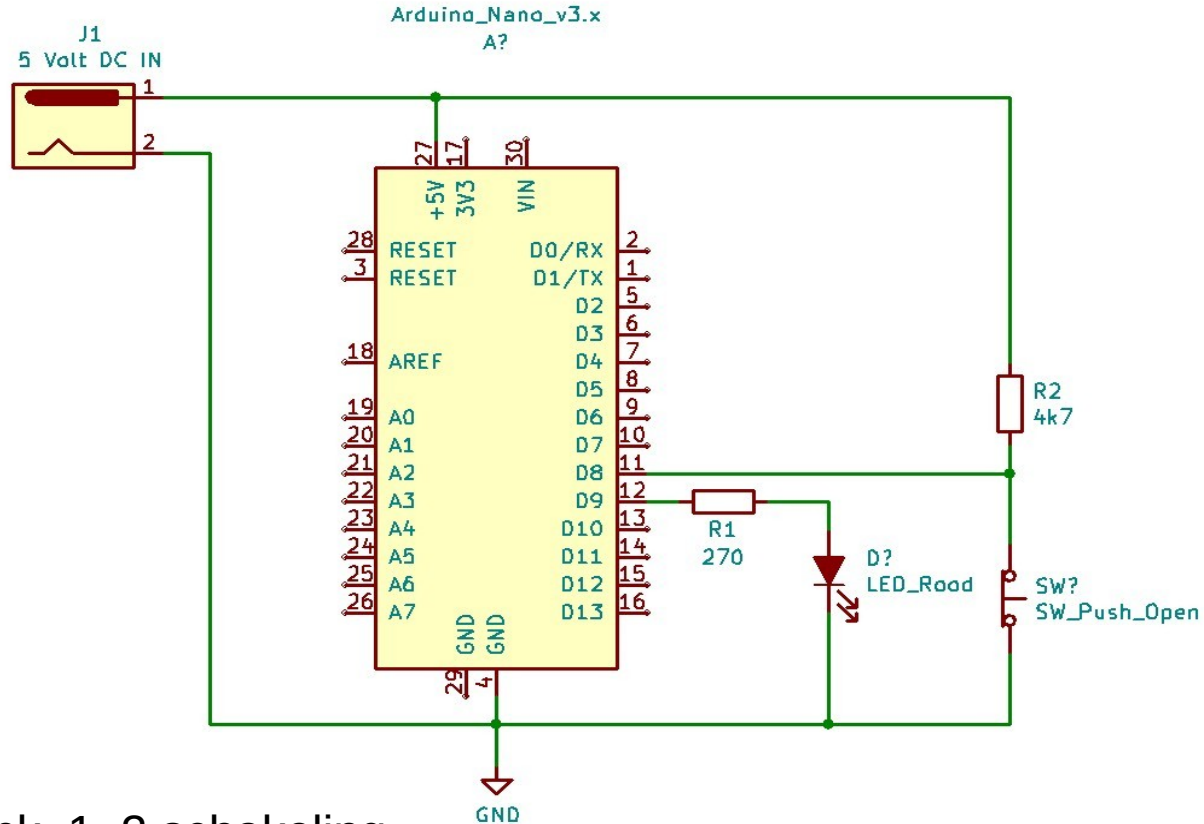
Opslaan voltooid.
avrdude done. Thank you.
```

4 Arduino Nano op /dev/ttyUSB0

Lezen en schrijven met digitale in en outputs.

Het programma hiernaast schakelt een led repeterend 1 seconde aan en 2 seconden uit afhankelijk van de stand van een schakelaar. We hebben dus een lees en een schrijf actie.

Arduino introductie.



Schema Blink_1_2 schakeling

Arduino introductie.

De functie van de digitale pinnen in setup instellen.

Digitale pinnen D0.... D13 als **output**: `pinMode` (0 tot 13, `OUTPUT`);

Digitale pinnen D0.... D13 als **input**: `pinMode` (0 tot 13, `INPUT`);

Voor 0 tot 13 kun je ook een eerder gedeclareerd alias invullen.

Voorbeeld: `int` `LedR` = 9; `pinMode` (`LedR`, `OUTPUT`);
`LedR` is een rode led die wordt aangesloten op pin D9.

Ieder statement wordt afgesloten met een punt komma.

Arduino introductie.

Digitale pinnen gebruiken in een toepassing.

`digitalWrite` (LedR, `HIGH`); // dit statement maakt pin LedR hoog (5Volt)

`digitalWrite` (LedR, `LOW`); // dit statement maakt pin LedR laag (0 Volt)

`digitalRead` (sense); // dit statement leest de booleaanse waarde (0 of 1)

Wil je de waarde van een leesactie gebruiken in het programma dan moet je die opslaan in een variabele die vooraf voor of in setup gedeclareerd moet zijn.

Voorbeeld: `val = digitalWrite` (sense);

De inhoud van de variabele `val` kun je vervolgens gebruiken in je programma bijvoorbeeld in een conditionele test.

Arduino introductie.

Pulse Width Modulation met de digitale pinnen.

Met PWM kun je de helderheid van een led, lamp of het toerental van een gelijkstroommotor regelen.

Bij de Arduino Uno en Nano kun je daarvoor pinnen 3, 5, 6, 9, 10, en 11 gebruiken.

Bij de pinnen 3, 9, 10 en 11 is de frequency 490 Hz en bij pinnen 5 en 6 980 Hz.

Het regelen gebeurt door de pulsbreedte te varieëren.

Voorbeeld: `int` LedR = 9; `int` val; val = 127;
`analogWrite`(LedR, val);

De output op pin D9 is een 490 Hz blokgolf met een dutycycle van 50%.

De waarde van val moet tussen 0 en 255 liggen. Als je voor de waarde van val gebruik maakt van een `analogRead` statement moet je de waarde door 4 delen. `analogRead` heeft een resolutie van 10 bits (0 tot 1023).

Arduino introductie.



The screenshot shows the Arduino IDE interface with the file 'Nano_pwm_01' open. The code is a C++ sketch for pulse width modulation on an Arduino Nano. It includes a multi-line comment explaining the demo, which uses the analogRead function to read a value from pin A0 and then uses analogWrite to output a PWM signal to pin D9. The setup function configures pin D9 as an output. The loop function reads the analog value and writes the PWM signal. Below the code editor, a status bar indicates 'Opslaan voltooid.' (Save complete.) and 'avrdude done. Thank you.' The bottom status bar shows '282' and 'Arduino Nano op /dev/ttyUSB0'.

```
Nano_pwm_01 | Arduino 1.8.11
Bestand Bewerken Schets Hulpmiddelen Help

Nano_pwm_01

/* Demo puls width modulation. Pulse width Modulation op D3,D5,D6,D9, D10 en D11 wordt geactiveerd
 * met het analogWrite statement. In dit voorbeeld verschijnt er een blokspanning met een vaste
 * frequency van 490 Hz op D9. De tijd dat de puls hoog is in verhouding tot de
 * tijdsduur van de hele periode kan gevarieerd worden. In het programma wordt dit
 * bepaald door de waarde van variabele val. Die waarde moet tussen 0 en 255 liggen.
 * De inhoud van variabele val is het resultaat van een analogRead actie op pin A0 (analogin0 en
 * zal tussen 0 en 1023 liggen (10 bits resolutie). De gemeten waarde moet worden gedeeld door
 * vier om in het bereik van 0 tot 255 te liggen.
 */
int val; // variabele val
int LedR = 9; // LedR alias voor pin D9
int analogin = A0; // analogin alias voor pin A0
void setup() {
  pinMode (LedR, OUTPUT); // LedR als output gedefinieerd
}

void loop() {
  val = analogRead(analogin); // de waarde op analogin wordt ingelezen in variabele val
  analogWrite(LedR, val/4); // de waarde in val wordt gedeeld door 4 en als sturing gebruikt
  // voor de PWM.
}
```

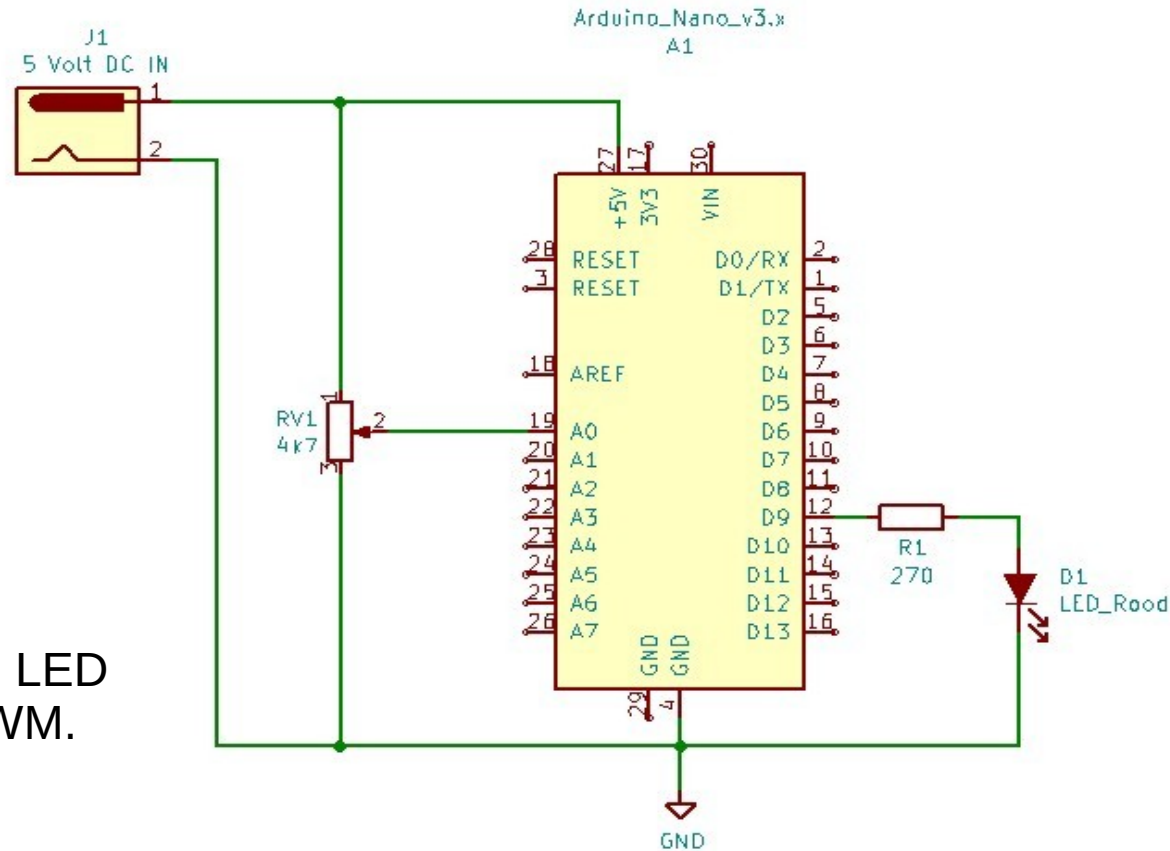
Opslaan voltooid.

avrdude done. Thank you.

282 Arduino Nano op /dev/ttyUSB0

Arduino introductie.

Helderheid van LED
regelen met PWM.



Arduino introductie.

De Arduino Uno en Nano hebben geen WiFi of ethernet aansluiting.
De consequentie hiervan is dat er ook geen actuele tijd is.

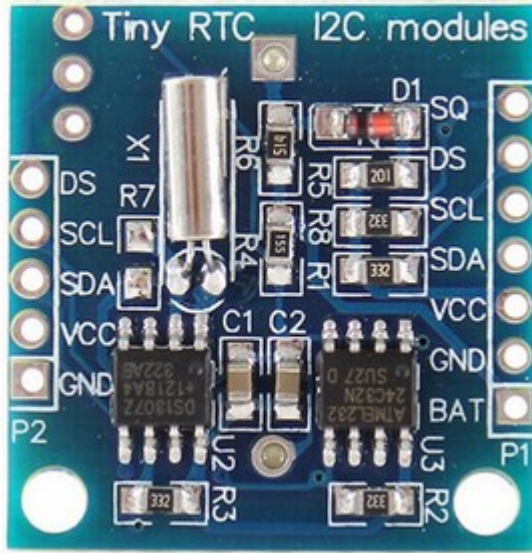
Voor veel toepassingen heb je een precieze tijd nodig.

Een **Real Time Clock (RTC)** kan hiervoor zorgen. Een RTC is een clock circuit met een backup batterij. Eenmaal op de juiste datum en tijd ingesteld kan het wel meer dan een jaar zonder externe spanning de juiste tijd en datum blijven geven.

Voor de Arduino kun je gebruik maken van de populaire DS1307 of de wat preciezere DS3231. De DS1307 kost tussen € 2.50 en 5.00
Beide boards communiceren met de Arduino via het I2C interface.
Om de DS1307 te kunnen gebruiken moeten de RTCLib.h en Wire.h bibliotheken geïnstalleerd zijn.

Deze bibliotheken kun je downloaden van de Arduino.org site.

Arduino introductie.



27 mm

De Tiny RTC module sluit je aan op de arduino via het I2C interface.

SDA wordt verbonden met pin A4, SCL met A5 op het Arduino Uno of Nano board.

Verder kun je naar keuze VCC op 3,3 of 5 Volt aansluiten.

Als laatste een verbinding met GND.

Op de volgende sheet zie je een kort programma waarmee je het I2C adres kunt vinden.

Arduino introductie.

```
#include <Wire.h>

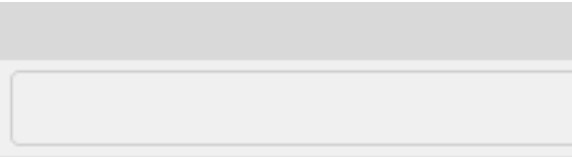
void setup() {
  Serial.begin (9600);
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;

  Wire.begin();
  for (byte i = 8; i < 120; i++)
  {
    Wire.beginTransmission (i);
    if (Wire.endTransmission () == 0)
    {
      Serial.print ("Found address: ");
      Serial.print (i, DEC);
      Serial.print (" (0x");
      Serial.print (i, HEX);
      Serial.println (");");
      count++;
      delay (1); // maybe unneeded?
    } // end of good response
  } // end of for loop
  Serial.println ("Done.");
  Serial.print ("Found ");
  Serial.print (count, DEC);
  Serial.println (" device(s).");
} // end of setup
```

Het programma test welke I2C adressen in gebruik zijn. De communicatie verloopt door middel van de wire.h bibliotheek.

De resultaten worden getoond op de Serial Monitor.

Controleer bij problemen of de Serial Monitor baud rate wel is ingesteld op 9600 baud.



```
I2C scanner. Scanning ...
Found address: 80 (0x50)
Found address: 104 (0x68)
Done.
Found 2 device(s).
```

Arduino introductie.

```
#include <Wire.h>
#include <RTClib.h>
RTC_DS1307 rtc;

void setup() {
  Serial.begin(9600);
  if(!rtc.begin()){
    Serial.println("RTC NOT WORKING");
    while (true);
  }
  rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
}

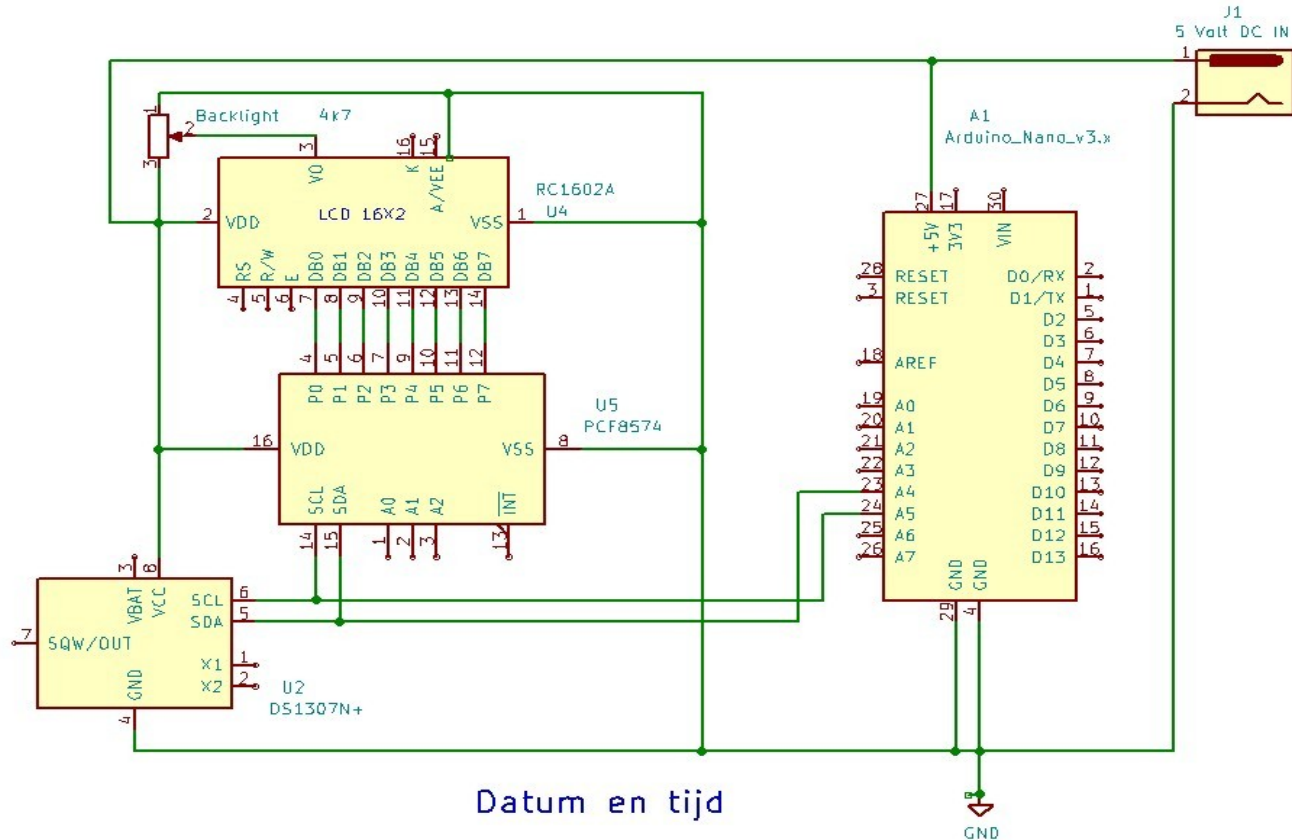
void loop() {
  DateTime now = rtc.now();
  Serial.print("Jaar: ");
  Serial.println(now.year(), DEC);
  Serial.print("Maand: ");
  Serial.println(now.month(), DEC);
  Serial.print("Dag: ");
  Serial.println(now.day(), DEC);
  Serial.print("Uur: ");
  Serial.println( now.hour() , DEC);
  Serial.print("Min: ");
  Serial.println(now.minute(), DEC);
  Serial.print("Seconde: ");
  Serial.println(now.second(), DEC);
  delay(5000);
  Serial.println("");
}
```

RTC.setup programma. Het programma laat de resultaten zien op de Serial Monitor. De RTC wordt tijdens het uitvoeren van de setup gesynchroniseerd met de clock van de laptop of PC mits de arduino daarmee is verbonden.

In de Serial Monitor zie je de actuele datum en de tijd. Elke 5 seconden wordt de data ververst.

Als in de toepassing het contact met de laptop verbroken is, kan er op den duur een afwijking ontstaan door een onnauwkeurigheid in de kristal frequentie en door temperatuur drift.

Arduino introductie.



Datum en tijd

Datum en tijd weergeven op een LC karakter Display,

Arduino introductie.

Datum en tijd op een karakter LCD.

```
lcd_tut_klok 5

#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // LCD library
#include <RTClib.h> // Real Time Clock library
RTC_DS1307 rtc; // rtc als alias voor RTC_DS1307
int uur; // variabele voor het uur van de dag
int minu; // variabele voor de minuut in het uur
int seco; // variabele voor de seconde in de minuut
int dag; // variabele voor de dag van de maand
int maand; // variabele voor de maand van het jaar
int jaar; // variabele voor het jaar getal

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // lcd als alias voor LiquidCrystal_I2C
// met vermelding van het I2C adres
// en het type LCD 2 rijen 16 karakters

void setup() {
  lcd.clear();
  rtc.begin();
  if (rtc.isrunning()) {
    lcd.print("RTC is not running");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // RTC synchroniseren met laptop clock
  }
  lcd.cursor(); // LCD initieeren
  lcd.init();
  lcd.backlight();
}
```

Initialisatie en setup.

Arduino introductie.

Datum en tijd elke 10 seconden naar LCD schrijven.

```
void loop(){
    DateTime now = rtc.now();
    dag = (int) now.day();
    maand = (int) now.month();
    jaar = (int) now.year();
    uur = (int) now.hour();
    minu = (int)now.minute();
    seco = (int)now.second();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print ("Datum: ");
    lcd.print(dag);
    lcd.print("/");
    lcd.print(maand);
    lcd.print("/");
    lcd.print(jaar);
    lcd.setCursor(0,1);
    lcd.print("Tijd:");
    lcd.setCursor(7,1);
    lcd.print(uur);
    lcd.print(":");
    lcd.print (minu);
    lcd.print(":");
    lcd.print(seco);
    delay(10000);
    // clock van Nano synchroniseren met RTC
    // dag getal 1....31 laden in variabele dag
    // maand getal laden in variabele maand
    // jaar getal laden in variabele jaar
    // uurtijd laden in variabele uur
    // minuut waarde laden in variabele minu
    // seconden getal laden in variabele seco
    // zet cursor op positie 1 van regel 1
    // print tekst Datum:en spatie op LCD
    // print dag getal op LCD
    // print backslash als scheidingsteken
    // print maand getal op LCD
    // print backslash als scheidingsteken
    // print jaar getal op LCD
    // zet cursor op positie 0 van regel 2
    // print tekst Tijd: op LCD
    // zet cursor op positie 7 van regel 2
    // print uur getal op LCD
    // print dubbele punt
    // print minuten getal op LCD
    // print dubbele punt
    // print seconden getal op LCD
    // wacht 10 seconden
```

Arduino introductie.

Documentatie.

<https://www.arduino.cc/en/main/docs>

<https://www.arduino.cc/reference>

http://bit.ly/eve_arduino

Arduino Cursus 2018 – Paul van Veen (Internet)
Arduino voor dummies. (Bol.com)

Arduino introductie.

De Arduino Nano kun je bij eenvoudige experimenten voeden vanuit de USB aansluiting van de laptop. Echter zodra je de verbinding verbreekt stopt de Arduino.

Als alternatief kun je gebruik maken van een oude telefoon adapter(kringloop winkel) of 4 oplaadbare batterijen. Vier oplaadbare batterijen in een houder levert $4 \times 1.2 = 4.8$ volt op, voldoende om de Arduino te voeden.

Tot slot voor de Arduino is er een groot aantal bibliotheken voor bijna elke toepassing beschikbaar op [Arduino.org](https://www.arduino.org)

Arduino introductie.

Wat heb je nodig om te starten?

- Arduino Uno of Nano. (Ben's Electronics of Hobby Electronica)
- Breadboard (de Nano kan hier direct worden in geplugged.
- Patch draden om verbindingen te maken.
- Een 5 volt voeding minimaal 400 mAmp. (Kringloop?)
- LED.
- Weerstand 270 ohm $\frac{1}{4}$ watt.
- Potentiometer 4k7 ohm

Arduino IDE downloaden van [Arduino.org](https://www.arduino.org)

- Kabeltje USB A naar USB Micro (Nano) of
- Kabeltje USB A naar USB B (UNO).

Arduino introductie.

Vragen?