

# SimulIDE simulatie programma – Korte handleing

## Installatie op Windows

- Download het zip bestand via <https://simulide.com/p/download110/>
- Verplaats de zip naar in een map met de naam SimulIDE op de C schijf.
- Pak de zip uit.

Nu staat er een map met de naam: SimulIDE\_1.1.0-SR2\_Win64.

In deze map staat: simulide.exe

Dat is het programma.

Met een dubbelklik op deze naam wordt het programma gestart.

## Handleiding

De handleiding is te vinden onder: <https://simulide.com/p/basic-use/>

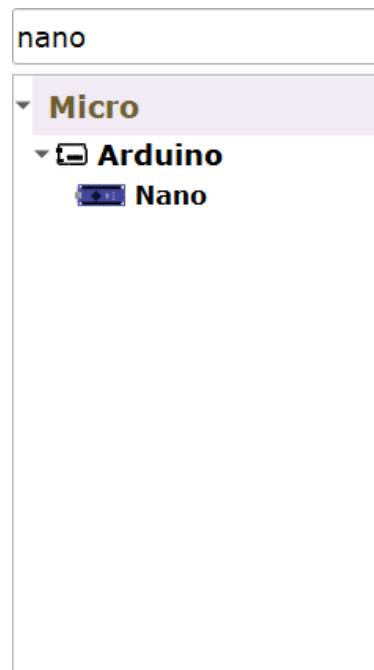
Daarnaast zijn er diverse tutorials te vinden op Youtube.

## Basis info

Het programma heeft geen save knop. Met sluiten via het rode kruis rechtsboven worden de wijzigingen automatisch opgeslagen.

## Onderdelen voor schakeling

In het linker deel van het scherm staat een kolom. Hier staan de onderdelen voor het samenstellen van een schema. Als voorbeeld hebben we een Arduino nano gezocht.



## Schakeling venster (Breadboard)

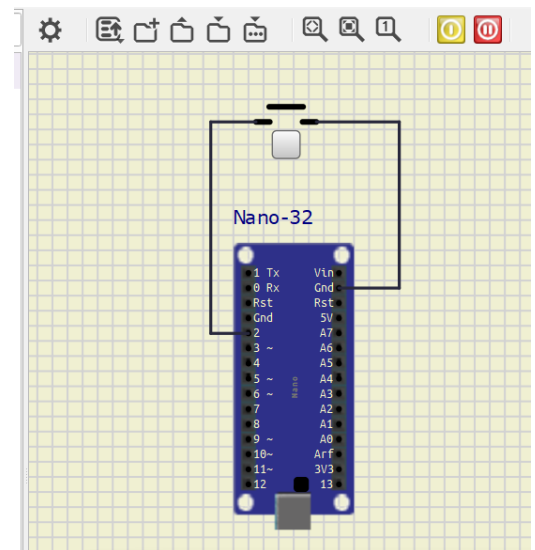
Het breadboard waar de schakeling op gebouwd kan worden staat daar naast aan de rechterkant. De onderdelen kun naar dit veld verslepen.

Dit veld heeft de volgende knoppen.



Van links naar rechts de werking:

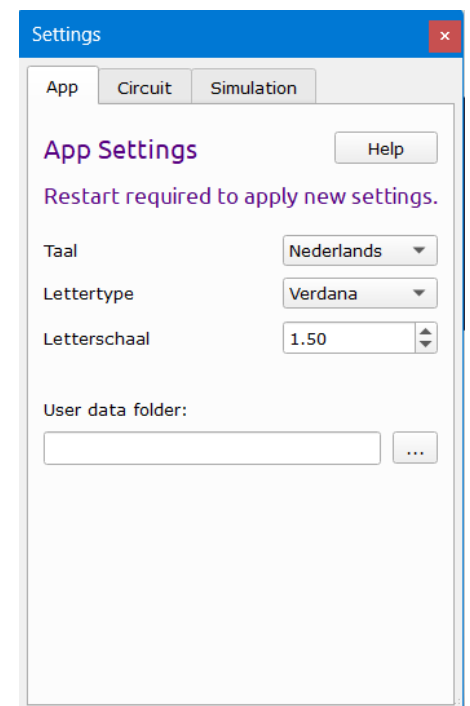
- Instellingen voor dit scherm
- Openen van een schakeling
- Nieuwe schakeling
- Bestaande schakeling openen
- Schakeling opslaan
- Schakeling opslaan als
- Stop simulatie
- Simulatie pauzeren



Van de knop settings is de eerste tap “App” interessant. Hier kan de taal, het lettertype en de schaal van het lettertype ingesteld worden. Tevens de plek waar standaard je circuits opgeslagen worden. “User data folder”.

Programma’s hebben de extensie: .ino.

Schakelingen hebben de extensie: .sim



## Programma venster

Het programma venster staat rechts van dit breadboard en heeft de volgende functie balk.



Van links naar rechts:

- Instellingen voor dit scherm
- Openen van een programma
- Bestaand programma openen
- Programma opslaan
- Programma opslaan als
- Zoek en vervang
- Compileren
- Uploaden naar de processor
- Start debug.

```
1
2
3 volatile byte state = LOW;
4
5 void blink();
6
7 void setup () {
8   Serial.begin(9600);
9   pinMode (13, OUTPUT);
10  pinMode (2, INPUT_PULLUP);
11  attachInterrupt(digitalPinToInterrupt(2), blink, CHANGE);
12  Serial.println ("start");
13  //interrupts();
14  }
15
16 void loop() {
17   digitalWrite(13, state);
18 }
19
20 void blink() {
21   state = !state;
22   Serial.println (state);
23 }
```

Onder instellingen heb je de volgende mogelijkheden:  
Editor settings, File settings en Compiler settings.

### Editor settings

Hier kan lettertype en tabgrootte aangepast worden.

### File settings

Hier moet de compiler Arduino gekozen worden. Daarnaast kunnen enkele standaard functie ingesteld worden zoals opslaan bij sluiten, laden van de compiler etc.  
Deze vinkjes zou ik allemaal aanzetten.

### Compiler settings

Onder deze tab moet de plaats gekozen worden waar het Arduino IDE staat. Als dat standaard geïnstalleerd is dan staat het onder program files.

Tool Path  ...

Kies dit pad.

Het zetten van breakpoints kan door het klikken op de cijfers van de programma regels. Uiteraard kan dit alleen op regels waar een instructie staat. Als debug is gestart verschijnt het volgende



menu.

Van links naar rechts:

- Zoek en vervang
- Stap
- Stap over
- Ga tot breakpoint
- pauzeer
- Start opnieuw
- Stop debug

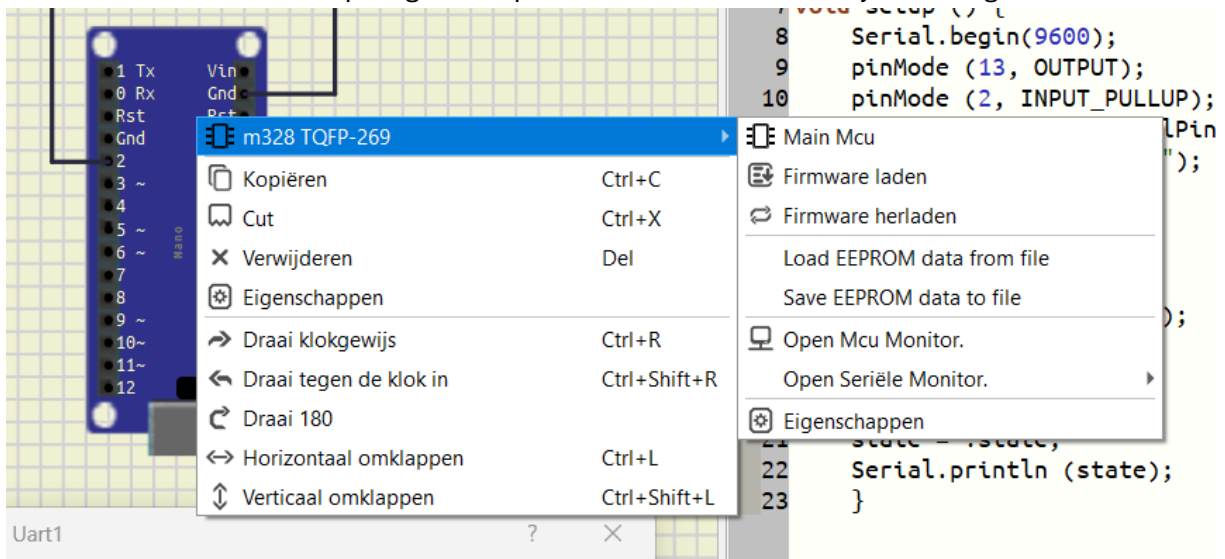
Bij het instellen van een breakpoint, door het klikken op een regelnummer, verschijnt er een gele blokje. Na het starten verschijnt op de eerste regel een pijltje. Vervolgens kan door het programma gestapt worden of direct naar het breakpoint gegaan worden.

Als je op het breakpoint bent aangekomen dan verschijnt het pijltje in het gele vlakje.

```
15
16 void loop() {
17     digitalWrite(13, state);
18 }
19
```

## Processor menu

Door met de rechtermuis op de gekozen processor te klikken verschijnt het volgende menu.



De eerste keuze heeft het processor menu waar je de monitor kunt open “Open Seriële Monitor.”. Hiermee wordt een venster geopend waar je de regels die met Serial.println in het programma te zien krijgt. Dit is belangrijk voor debugging.

## Monitor scherm

