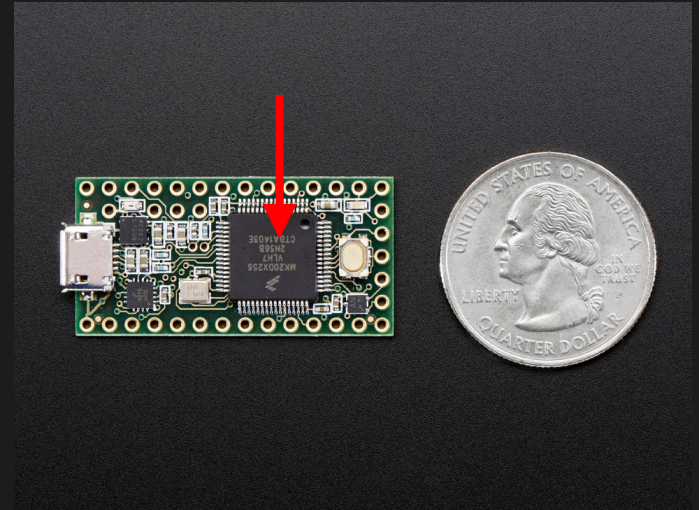


Hardware

Introductie Teensy

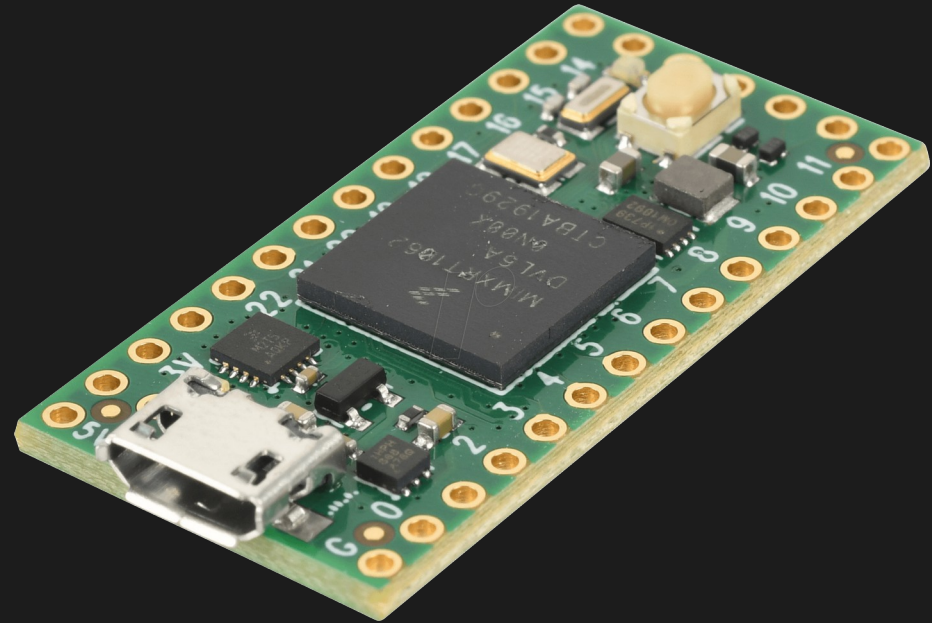
Microcontrollers

- Programmeerbare chip
- Interactie met sensoren:
 - Afstand / beweging
 - Temperatuur / vochtigheid
 - ...
- Actuators aansturen:
 - LEDs
 - Motors
 - Relais
 - ...



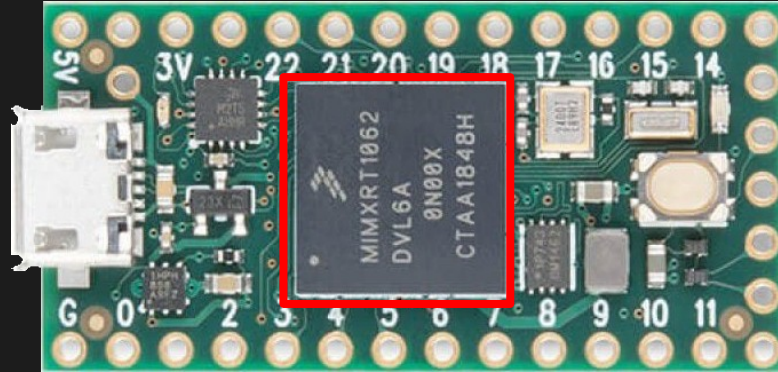
Teensy

- Microcontroller bord populair voor audio projecten:
 - Krachtige chip
 - USB MIDI
 - Audio library
- Ideaal als brug tussen software en de fysieke wereld

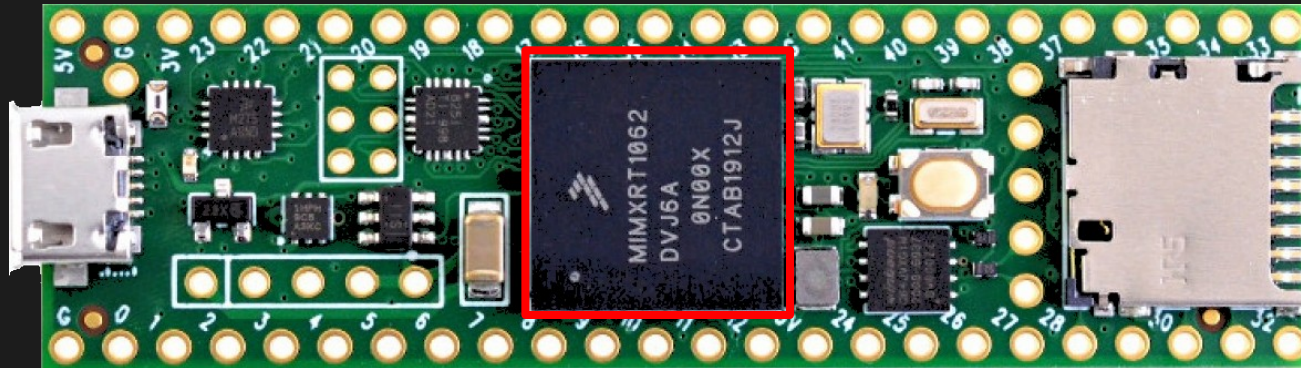


Teensy microcontroller

Teensy 4.0



Teensy 4.1



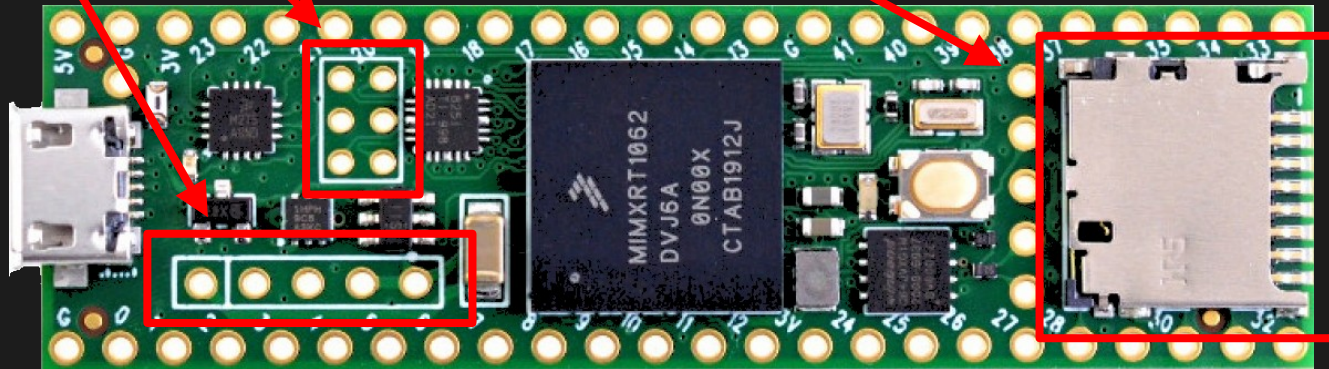
Teensy USB

- Micro USB voor:
 - Stroomvoorziening
 - Uploaden van programma



Teensy 4.1

- Extra functies:
 - Meer pinnen
 - Micro SD kaart lezer
 - Ethernet
 - USB-host



Teensy pins

Teensy 4.0



Teensy 4.1



Pins: power

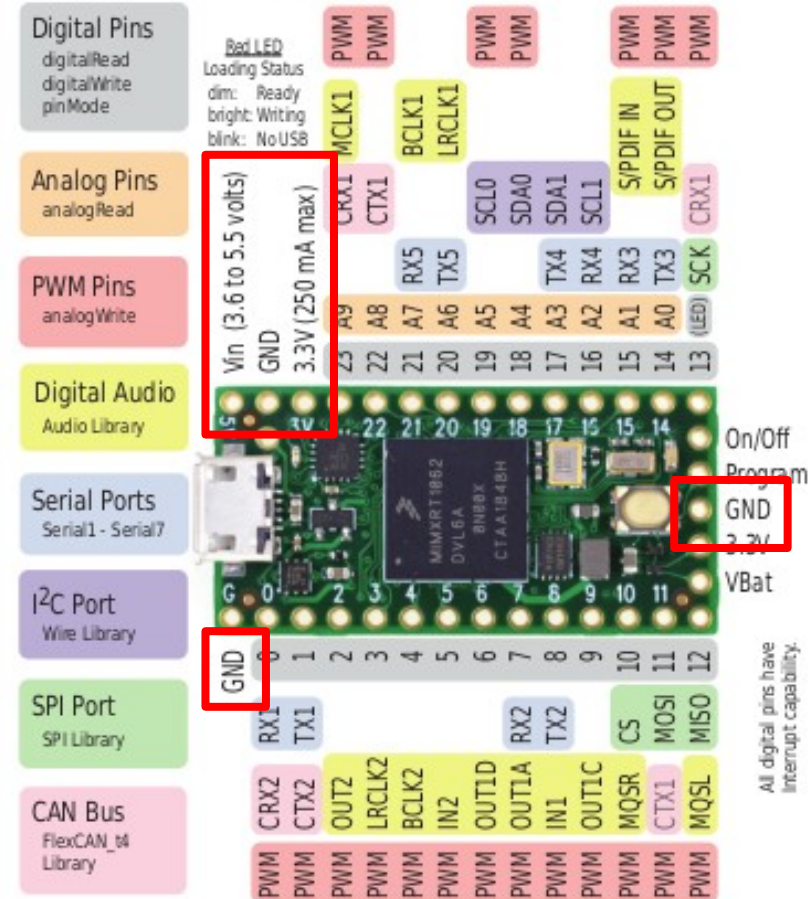
- Vin (~5V)
 - Verbonden met 5V van USB-kabel of te gebruiken als aansluiting voor een externe voeding
- 3.3V
 - Voedingsbron voor bijv. sensoren
- GND (0V)
- Meeste pinnen kunnen maximaal 3.3V aan!

Welcome to Teensy® 4.0

32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy



Pins: digital in/out

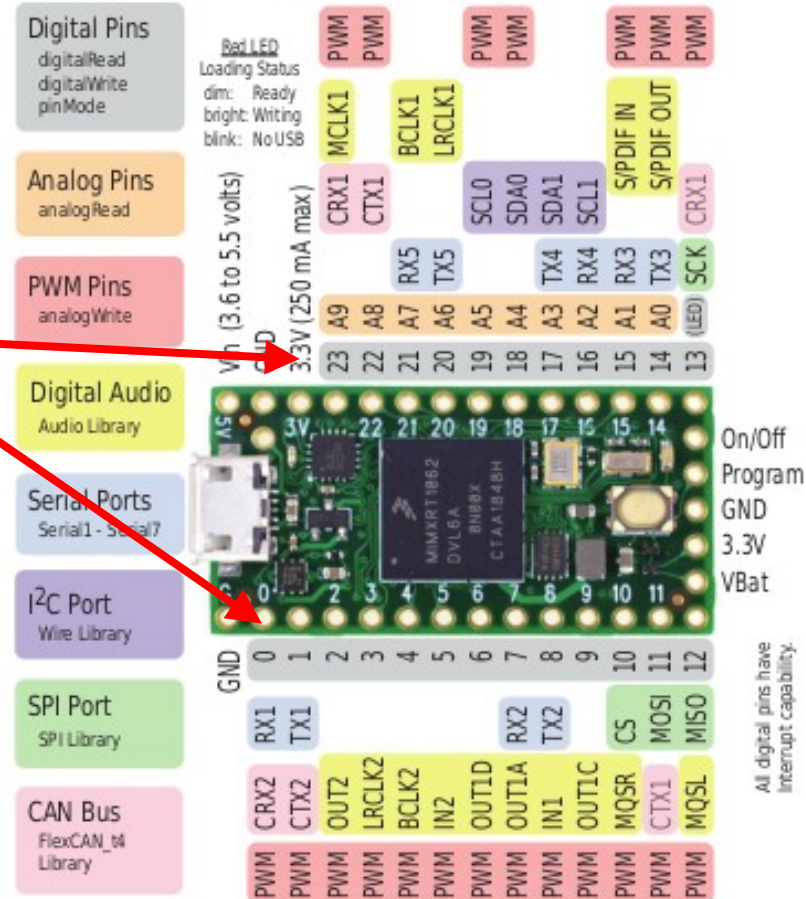
- Binair: 0V of 3.3V
 - 0V LOW 0 false
 - 3.3V HIGH 1 true
- Meeste pins digitaal inzetbaar
- Te gebruiken voor
 - Schakelaars
 - LEDs
 - Relais
 - ...

Welcome to Teensy® 4.0

32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy



Pins: analog in

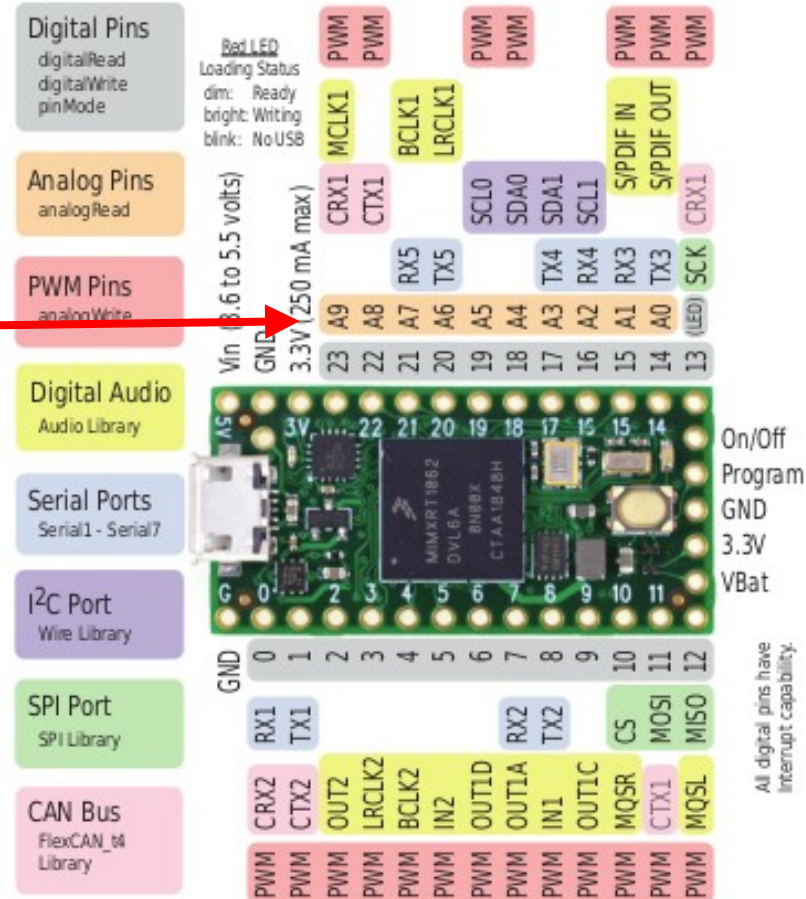
- Meet voltages tussen 0V en 3.3V
 - 10-bits: getallen tussen 0 en 1023
- Niet alle pinnen kunnen dit
- Te gebruiken voor
 - Potmeters
 - Diverse sensoren
 - ...

Welcome to Teensy® 4.0

32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy



Pins: 'analog' out

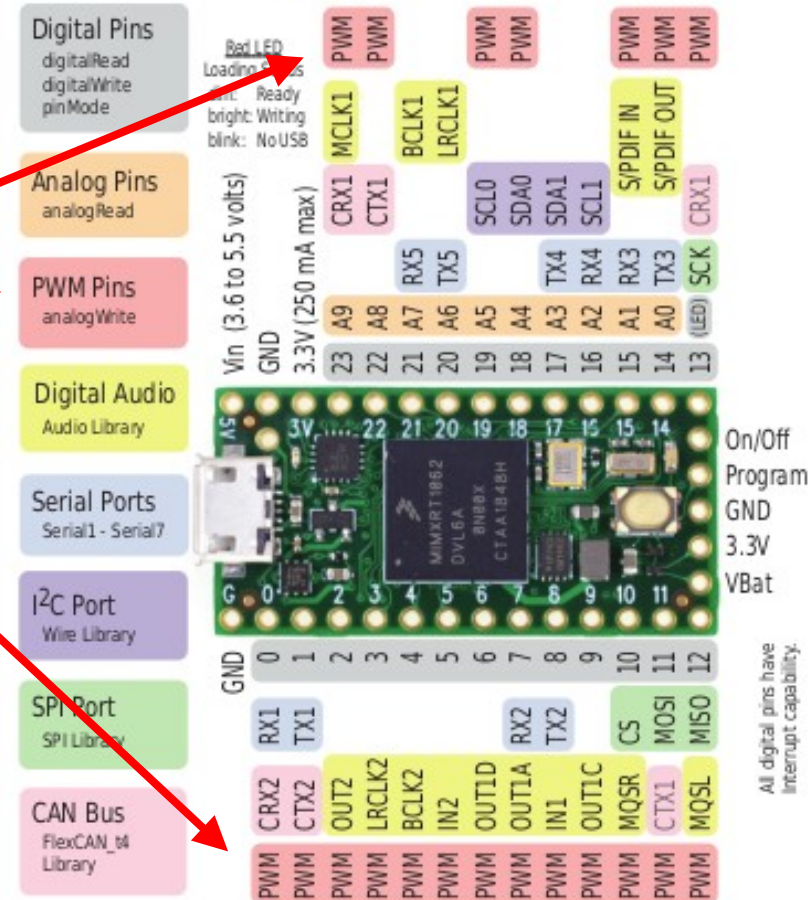
- Pulse-width modulation
 - Alternatief voor dure analoge outputs (DAC's)
- Veel pinnen kunnen dit
- Te gebruiken voor:
 - Dimmen van LEDs
 - Aansturen motoren
 - CV output voor modulaire synths
 - ...

Welcome to Teensy® 4.0

32 Bit Arduino-Compatible Microcontroller

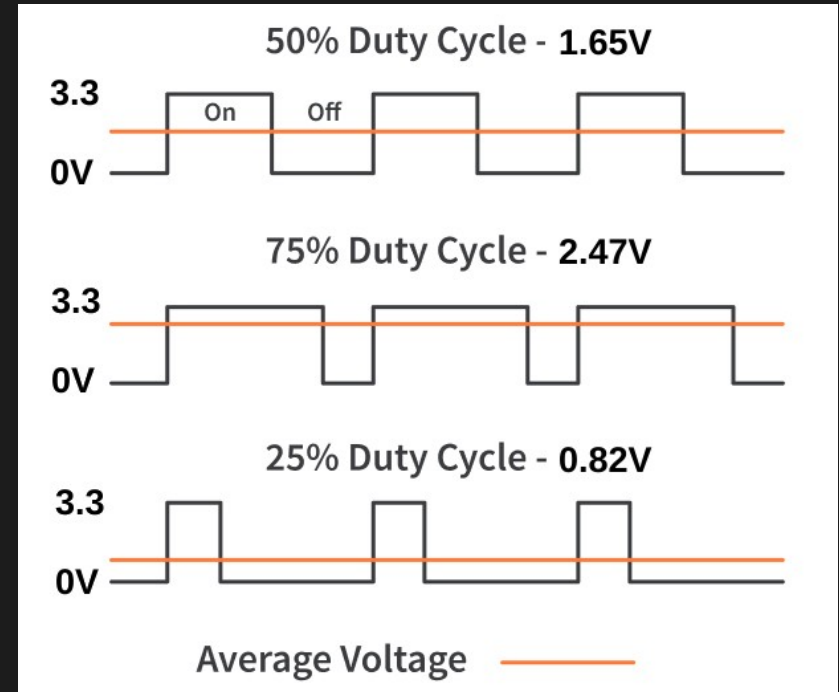
To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy



PWM

- Snel schakelen van een digitale pin
- Pulse breedte instelbaar van 0-255
 - Lage waarde = langer op 0V dan op 3.3V
 - Hoge waarde = langer op 3.3V dan op 0V



Pins: andere functies

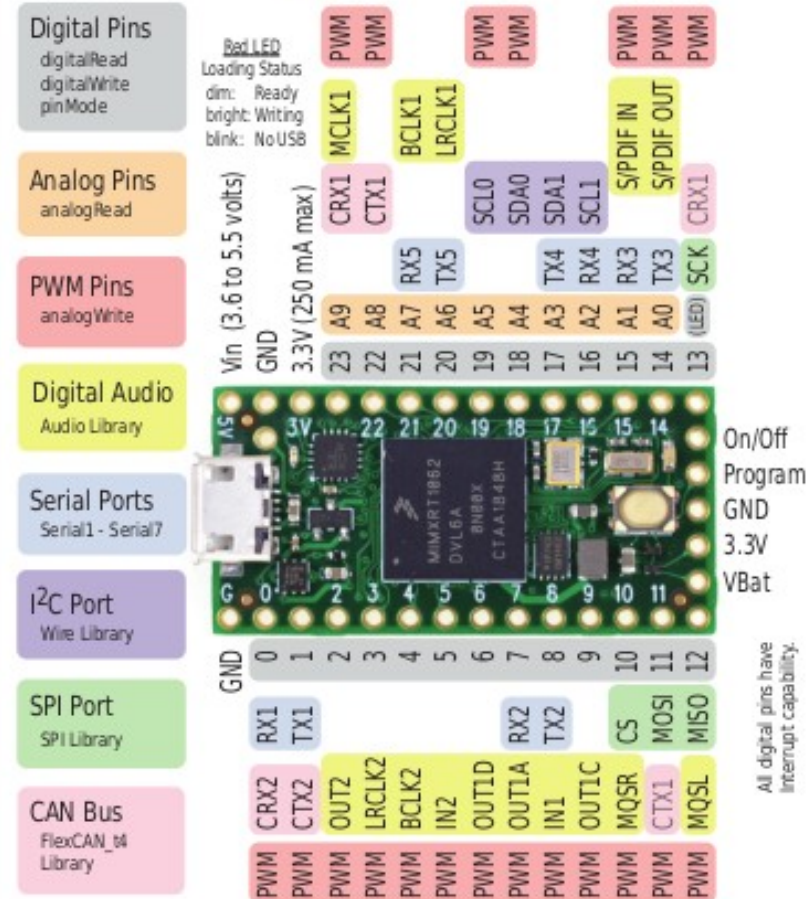
- Audio ADC/DAC aansturing
- Seriële communicatie
 - Bluetooth / Wi-Fi modules
 - MIDI naar andere hardware
- Andere protocollen
 - Complexe sensoren
 - Displays

Welcome to Teensy® 4.0

32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy

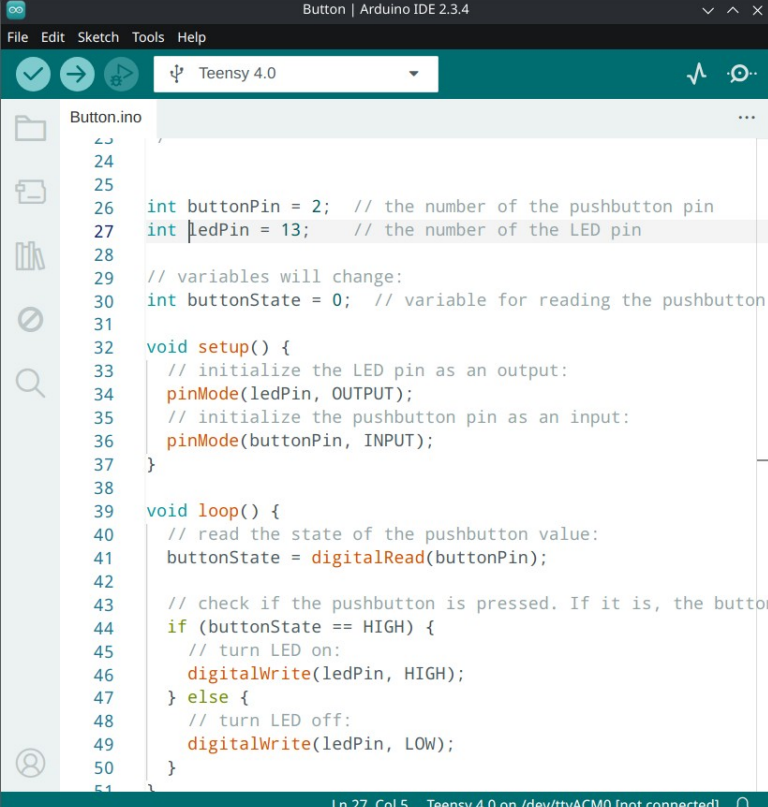


Hardware

Introductie Arduino

Arduino

- Programmeer omgeving voor microcontrollers
- Arduino IDE
 - Code editor
 - Compiler
 - Uploader/flasher van MCU's
 - Seriële monitor + plotter
- Lijkt op p5.js:
 - `void setup()`
 - `void loop()`



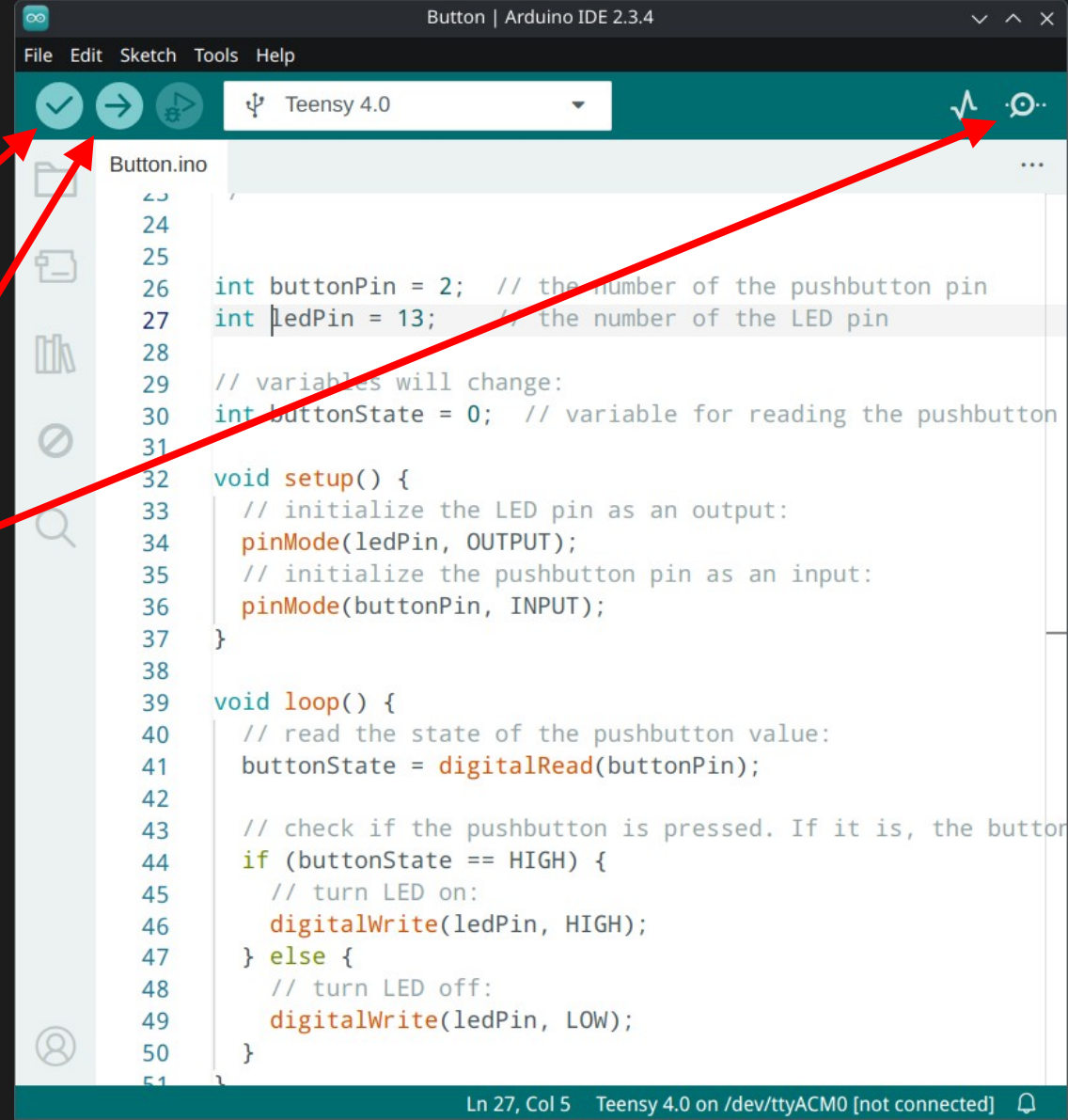
The screenshot shows the Arduino IDE 2.3.4 window. The title bar reads "Button | Arduino IDE 2.3.4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, and uploading, along with a dropdown menu set to "Teensy 4.0". On the left is a file explorer showing a folder named "Button.ino". The main editor area displays the following C++ code:

```
23 //
24
25
26 int buttonPin = 2; // the number of the pushbutton pin
27 int ledPin = 13; // the number of the LED pin
28
29 // variables will change:
30 int buttonState = 0; // variable for reading the pushbutton
31
32 void setup() {
33   // initialize the LED pin as an output:
34   pinMode(ledPin, OUTPUT);
35   // initialize the pushbutton pin as an input:
36   pinMode(buttonPin, INPUT);
37 }
38
39 void loop() {
40   // read the state of the pushbutton value:
41   buttonState = digitalRead(buttonPin);
42
43   // check if the pushbutton is pressed. If it is, the button
44   if (buttonState == HIGH) {
45     // turn LED on:
46     digitalWrite(ledPin, HIGH);
47   } else {
48     // turn LED off:
49     digitalWrite(ledPin, LOW);
50   }
51 }
```

The status bar at the bottom indicates "Ln 27, Col 5 Teensy 4.0 on /dev/ttyACM0 [not connected]" with a refresh icon.

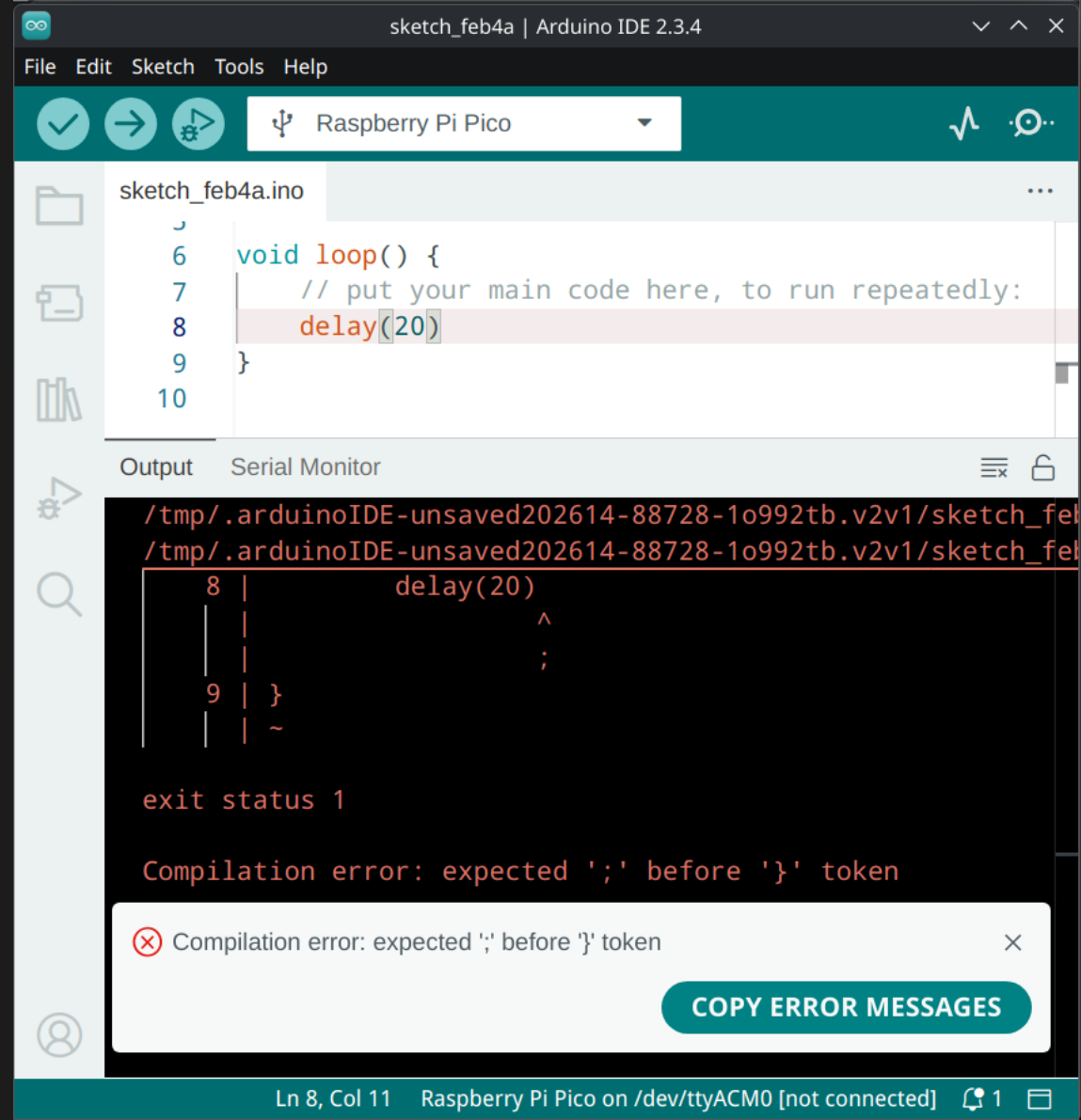
Arduino IDE

- Compileren/checken van je code
- Compileren/uploaden naar Teensy
- Seriële monitor
 - Hiermee kan je geprinte berichten vanuit je Teensy zien



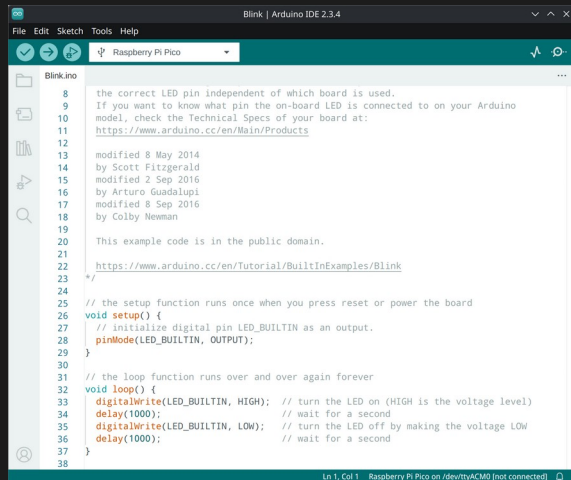
Arduino IDE

- Compileren gaat goed óf fout
- Output window:
 - Voortgang van compileren/uploaden
 - Foutmelding tijdens compilatie of upload



Arduino + Teensy

Code schrijven



```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

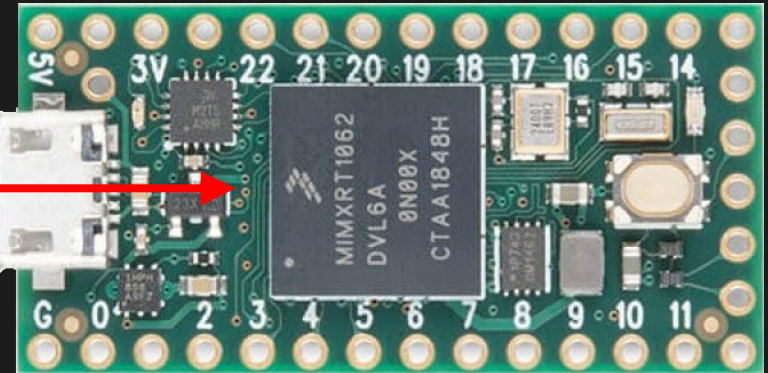
https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Uploaden

Zelfstandig code uitvoeren
(zonder tussenkomst van
computer of Arduino IDE)



Verschillen JS en Arduino / C++

- Draait 'interpreted' in browser
- Gebruik ; is wenselijk
- Variabele kan allerlei types bevatten
- Printen naar console:
 - `console.log("abc");`
- Draait gecompileerd op microcontroller
- Gebruik ; is verplicht
- Variabele is één specifiek vooraf bepaald datatype
- Printen over USB
 - `Serial.begin(9600);`
 - `Serial.println("abc");`

Verschillen JS en Arduino / C++

- `function setup()`
 - Eenmaals uitgevoerd zodra programma start
- `function draw()`
 - Uitgevoerd op de framerate van je browser/beeldscherm
- `void setup()`
 - Eenmaals uitgevoerd zodra programma start
- `void loop()`
 - Continue zo snel mogelijk uitgevoerd: `loop()` klaar? Gelijk nog een keer!

Datatypes

- Bij het aanmaken van een variabele bepaal je het type:
 - `int` van `-2147483648` tot `2147483647`
 - `float` van `-3.40282347E+38` tot `3.40282347E+38`
 - `bool` `true` of `false`
- Voorbeeld:
`float filterFreq = 530.76;`

Basis functies

- `pinMode(pin, mode);`
 - Stelt werking van pin in
 - Meestal in `setup()` gebruikt
 - Mode kan `INPUT`, `INPUT_PULLUP` of `OUTPUT` zijn
- `delay(value);`
 - Wacht aantal milliseconden

Basis functies: digitaal

- `digitalWrite(pin, value);`
 - Zet pin naar 0V of 3.3V
- `digitalRead(pin);`
 - Meet voltage op pin en geeft waarde terug
 - `0` als voltage 0V is
 - `1` als voltage 3.3V is

Basis functies: analoog

- `analogWrite(pin, value);`
 - Zet PWM aan en stelt breedte in op pin
 - `0` = 0% pulse breedte
 - `255` = 100% pulse breedte
- `analogRead(pin);`
 - Meet voltage op pin en geeft waarde terug
 - Waarde is tussen `0` (0V) en `1023` (3.3V)

Demo

- LED aanzetten
- Elke seconde iets printen