

Teensy les 1

# Teensy = Microcontroller board



# Teensy = Microcontroller board

interface tussen de analoge wereld en de digitale wereld

sensor-data lezen

[beweging, temperatuur, vocht, aanraking, etc.]

licht en beweging aansturen

[led, motor, relais]

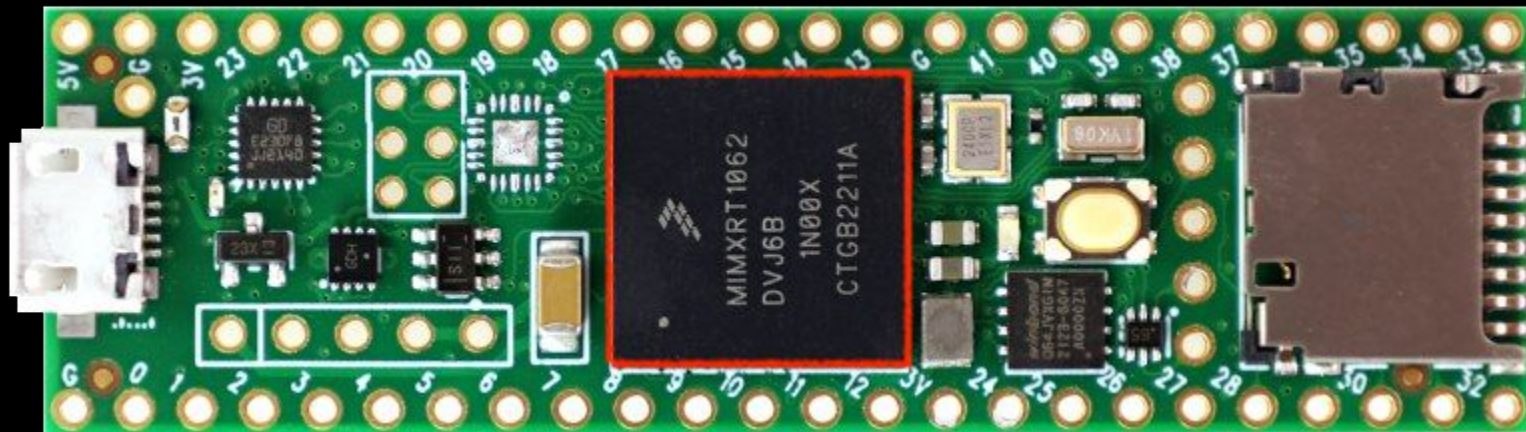
# Teensy = Microcontroller board

interface tussen de analoge wereld en de digitale wereld

gedrag van deze interface kan je programmeren

analoge wereld verbind je met de pinnen

# microcontroller





# (micro)USB-poort computer <-> Teensy

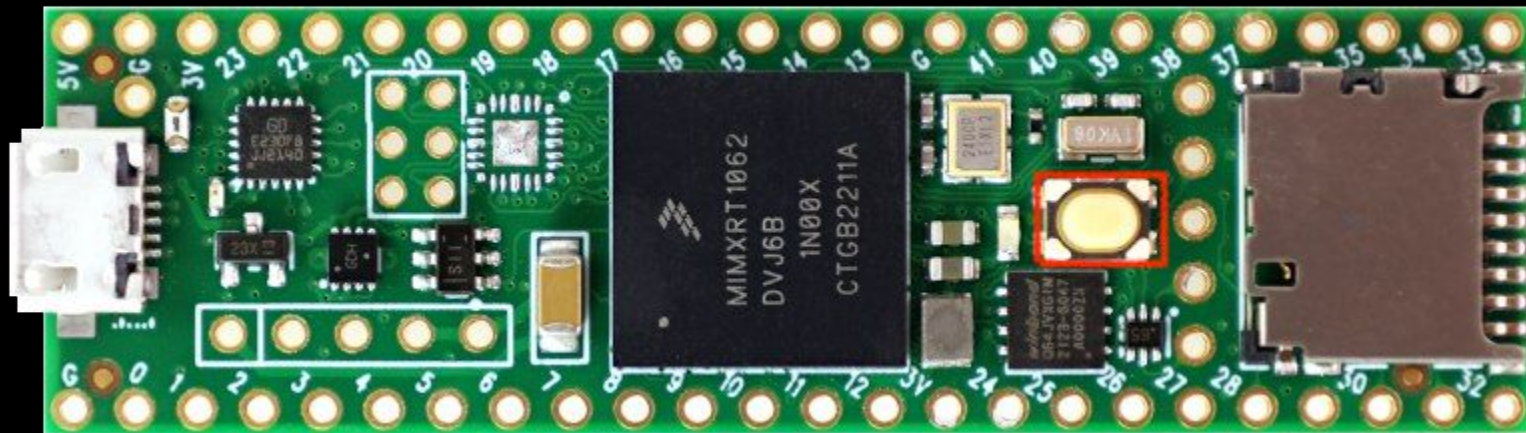


# Teensy workflow



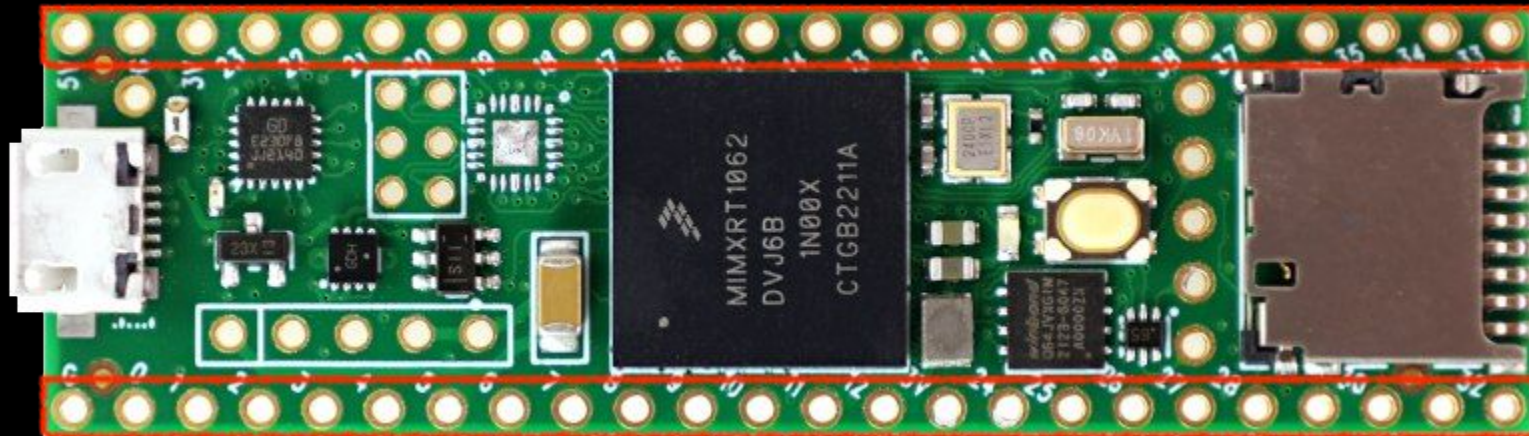
- Gecompileerde code vanuit Arduino IDE
- Seriële communicatie van en naar Teensy

# program button





pins



# 4.1 extra functies



micro-SD-kaart



ethernet



usb-host



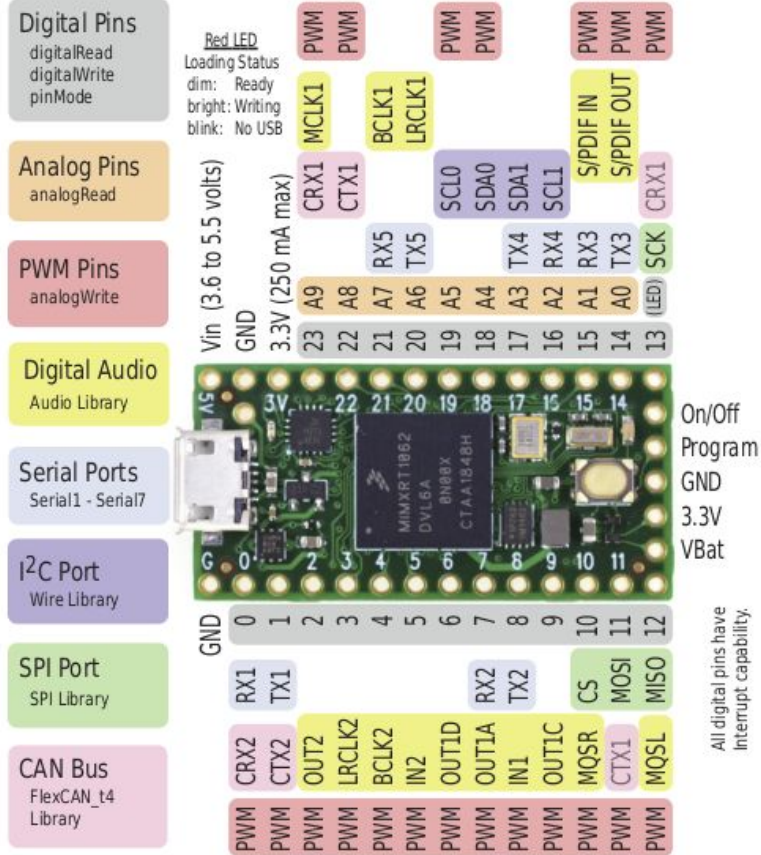
# alle pins

## 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](http://www.pjrc.com/teensy)

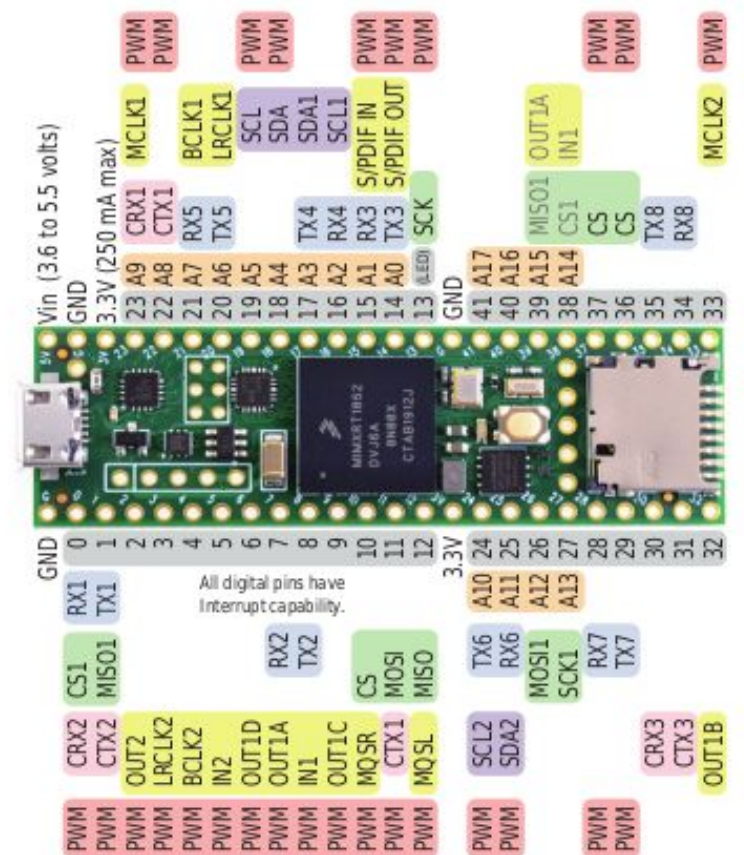


## Welcome to Teensy® 4.1

32 Bit Arduino-Compatible Microcontroller

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

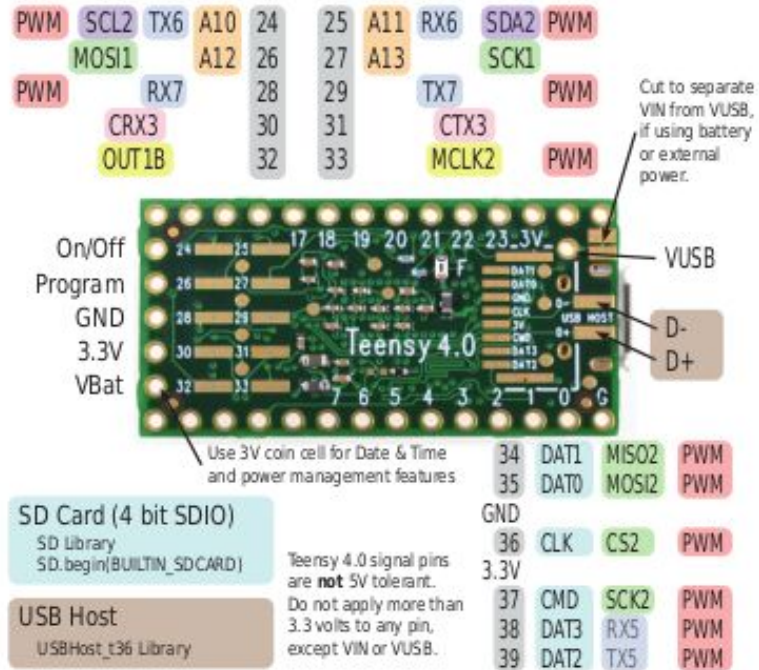
[www.pjrc.com/teensy](http://www.pjrc.com/teensy)



# alle pins

## Teensy® 4.0 Back Side

Additional pins and features available on the back side



**SD Card (4 bit SDIO)**  
SD Library  
SD.begin(BUILTIN\_SDCARD)

**USB Host**  
USBHost\_t36 Library

For solutions to the most common issues and technical support, please visit:

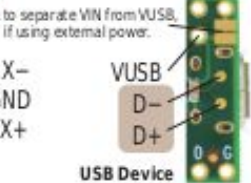
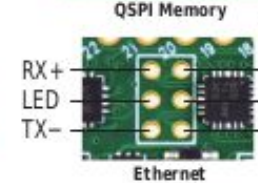
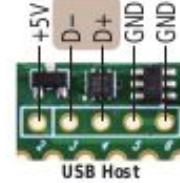
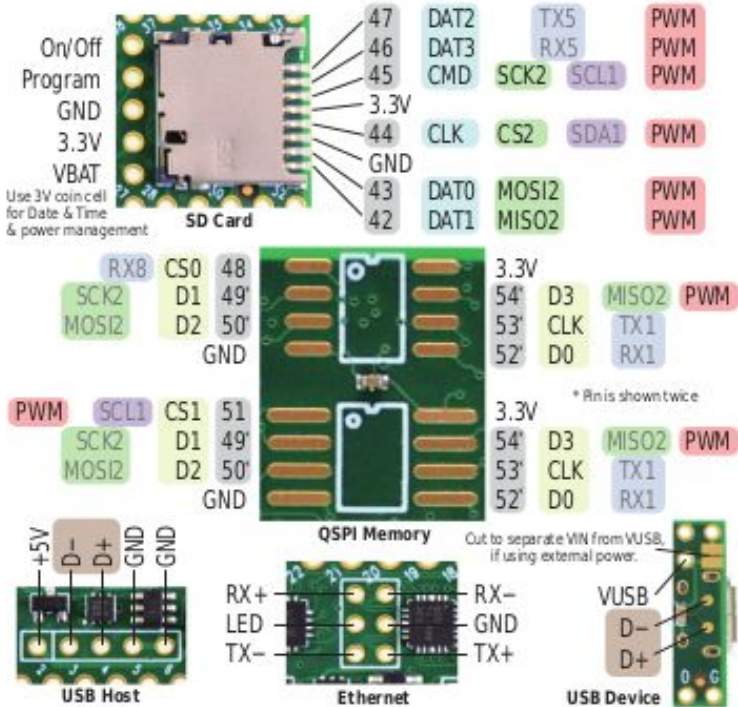
[www.pjrc.com/help](http://www.pjrc.com/help)

Teensy 4.0 System Requirements:  
PC computer with Windows 7, 8, 10 or later  
or Ubuntu Linux 14.04 or later  
or Macintosh OS-X 10.8 or later  
USB Micro-B Cable



## Teensy® 4.1 Features

Special Features and Additional Pins



For solutions to the most common issues and technical support, please visit:

[www.pjrc.com/help](http://www.pjrc.com/help)

Teensy 4.1 System Requirements:  
PC computer with Windows 7, 8, 10, 11 or later  
or Ubuntu Linux 14.04 or later  
or Macintosh MacOS 10.10 or later  
USB Micro-B Cable





# Power

meestal voeding via USB  
(5V)

Alle inputs en outputs  
kunnen maximaal 3.3V aan

met de 3.3V en GND-pins  
voed je sensoren e.d.

(VIN kan je gebruiken voor  
een externe voeding)

**Welcome to Teensy® 4.0**  
32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](http://www.pjrc.com/teensy).  
[www.pjrc.com/teensy](http://www.pjrc.com/teensy)

The diagram shows the Teensy 4.0 board with various pins and their functions. The pins are color-coded and labeled with their functions and the libraries they use. The board is shown with a USB-C port on the left and a micro-USB port on the right. The pins are numbered 0 to 23. The diagram includes the following categories:

- Digital Pins:** digitalRead, digitalWrite, pinMode. Pins include MCLK1, BCLK1, LRCLK1, S/PDIF IN, S/PDIF OUT, and several PWM pins.
- Analog Pins:** analogRead. Pins include CRX1, CTX1, RX5, TX5, TX4, RX4, RX3, TX3, and SCK.
- PWM Pins:** analogWrite. Pins include CRX1, CTX1, RX5, TX5, TX4, RX4, RX3, TX3, and SCK.
- Digital Audio:** Audio Library. Pins include RX1, TX1, RX2, TX2, CS, MOSI, MISO, and several PWM pins.
- Serial Ports:** Serial1 - Serial7. Pins include RX1, TX1, RX2, TX2, CS, MOSI, MISO, and several PWM pins.
- I<sup>2</sup>C Port:** Wire Library. Pins include RX1, TX1, RX2, TX2, CS, MOSI, MISO, and several PWM pins.
- SPI Port:** SPI Library. Pins include RX1, TX1, RX2, TX2, CS, MOSI, MISO, and several PWM pins.
- CAN Bus:** FlexCAN\_t4 Library. Pins include RX1, TX1, RX2, TX2, CS, MOSI, MISO, and several PWM pins.

Power pins are highlighted in red: VIN (3.6 to 5.5 volts), GND, 3.3V (250 mA max), and VBat. The 3.3V pin is also labeled as GND. The VBat pin is labeled as On/Off Program. A note at the bottom right states: "All digital pins have interrupt capability."

# Digitale in/outputs

**Welcome to Teensy® 4.0**  
32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](http://www.pjrc.com/teensy).  
[www.pjrc.com/teensy](http://www.pjrc.com/teensy)

The diagram shows the pinout of the Teensy 4.0 microcontroller. It includes a central image of the board with pins numbered 0 to 23. The pins are color-coded and labeled with their functions and associated libraries. A red box highlights pins 0 through 12, which are digital pins. A yellow box highlights pins 13 through 23, which are analog pins. The diagram also includes a table of pin functions and libraries on the left side.

Pin	Function	Library
0	Digital	digitalRead, digitalWrite, pinMode
1	Digital	digitalRead, digitalWrite, pinMode
2	Digital	digitalRead, digitalWrite, pinMode
3	Digital	digitalRead, digitalWrite, pinMode
4	Digital	digitalRead, digitalWrite, pinMode
5	Digital	digitalRead, digitalWrite, pinMode
6	Digital	digitalRead, digitalWrite, pinMode
7	Digital	digitalRead, digitalWrite, pinMode
8	Digital	digitalRead, digitalWrite, pinMode
9	Digital	digitalRead, digitalWrite, pinMode
10	Digital	digitalRead, digitalWrite, pinMode
11	Digital	digitalRead, digitalWrite, pinMode
12	Digital	digitalRead, digitalWrite, pinMode
13	Analog	analogRead
14	Analog	analogRead
15	Analog	analogRead
16	Analog	analogRead
17	Analog	analogRead
18	Analog	analogRead
19	Analog	analogRead
20	Analog	analogRead
21	Analog	analogRead
22	Analog	analogRead
23	Analog	analogRead

Min (3.6 to 5.5 volts)  
3.3V (250 mA max)

On/Off Program  
GND  
3.3V  
VBat

All digital pins have interrupt capability.

Aan of uit  
(3.3V of 0.0V)

vertaalt naar 1 of 0 of  
HIGH of LOW

input of output

schakelaar, led, relais

# Analoge inputs

**Welcome to Teensy® 4.0**  
32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](http://www.pjrc.com/teensy).  
[www.pjrc.com/teensy](http://www.pjrc.com/teensy)

The diagram shows the pinout of the Teensy 4.0 microcontroller. It is divided into several functional groups:

- Digital Pins:** digitalRead, digitalWrite, pinMode. Pins include MCLK1, BCLK1, LRCLK1, SCL0, SDA0, SDA1, SCL1, S/PDIF IN, S/PDIF OUT, CRX1, RX5, TX5, TX4, RX4, RX3, TX3, SCK, A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, GND, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, RX1, TX1, RX2, TX2, CS, MOSI, MISO, CRX2, CTX2, OUT2, LRCLK2, BCLK2, IN2, OUT1D, OUT1A, OUT1B, IN1, OUT1C, MQSR, CTX1, MQSL, RX1, TX1, RX2, TX2, CS, MOSI, MISO, CRX1, CTX1, OUT1, LRCLK1, BCLK1, IN1, OUT1A, OUT1B, IN1, OUT1C, MQSR, CTX1, MQSL.
- Analog Pins:** analogRead. Pins include CRX1, CTX1, SCL0, SDA0, SDA1, SCL1, S/PDIF IN, S/PDIF OUT, CRX1.
- PWM Pins:** analogWrite. Pins include MCLK1, BCLK1, LRCLK1, SCL0, SDA0, SDA1, SCL1, S/PDIF IN, S/PDIF OUT, CRX1, RX5, TX5, TX4, RX4, RX3, TX3, SCK, A0, A1, A2, A3, A4, A5, A6, A7, A8, A9.
- Digital Audio:** Audio Library. Pins include A0, A1, A2, A3, A4, A5, A6, A7, A8, A9.
- Serial Ports:** Serial1 - Serial7. Pins include TX5, RX5, TX4, RX4, TX3, RX3, TX2, RX2, TX1, RX1.
- I<sup>2</sup>C Port:** Wire Library. Pins include SCL0, SDA0, SDA1, SCL1.
- SPI Port:** SPI Library. Pins include SCL0, SDA0, SDA1, SCL1, CS, MOSI, MISO.
- CAN Bus:** FlexCAN\_t4 Library. Pins include CRX1, CTX1, OUT1, LRCLK1, BCLK1, IN1, OUT1A, OUT1B, IN1, OUT1C, MQSR, CTX1, MQSL.

Additional information: Min (3.6 to 5.5 volts), 3.3V (250 mA max), On/Off Program, GND, 3.3V, VBat. All digital pins have interrupt capability.

continu signaal  
tussen 0V en 3.3V

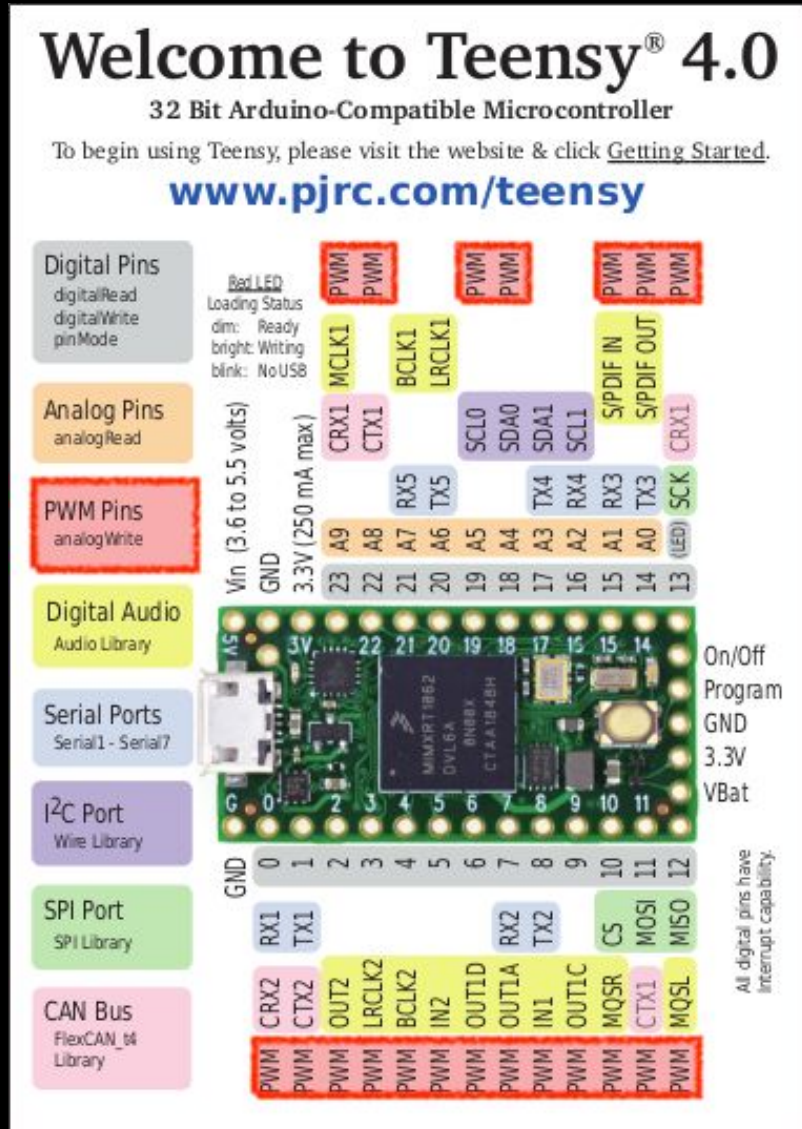
Vertaalt naar 10 bits getal  
0 - 1023

Alleen input

potmeter, afstandssensor,  
lichtsensor e.d.



# 'Analoge' outputs (PWM)



continu signaal  
(tussen 0V en 3.3V)

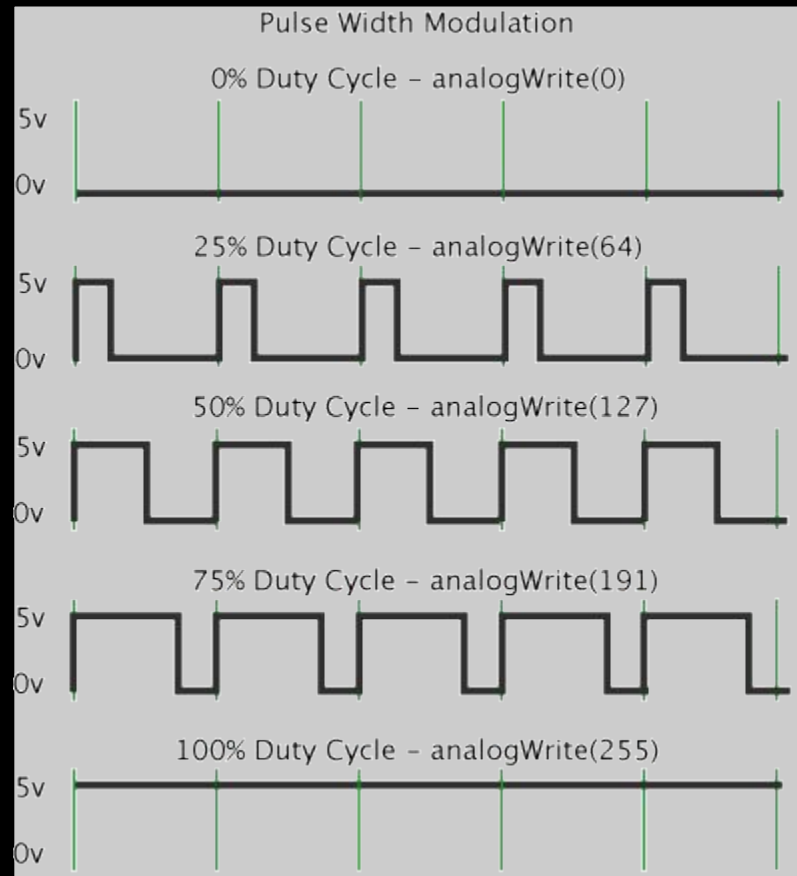
Vertaalt 0 - 1023 via Pulse Width Modulation naar 0V - 3.3V

Alleen output

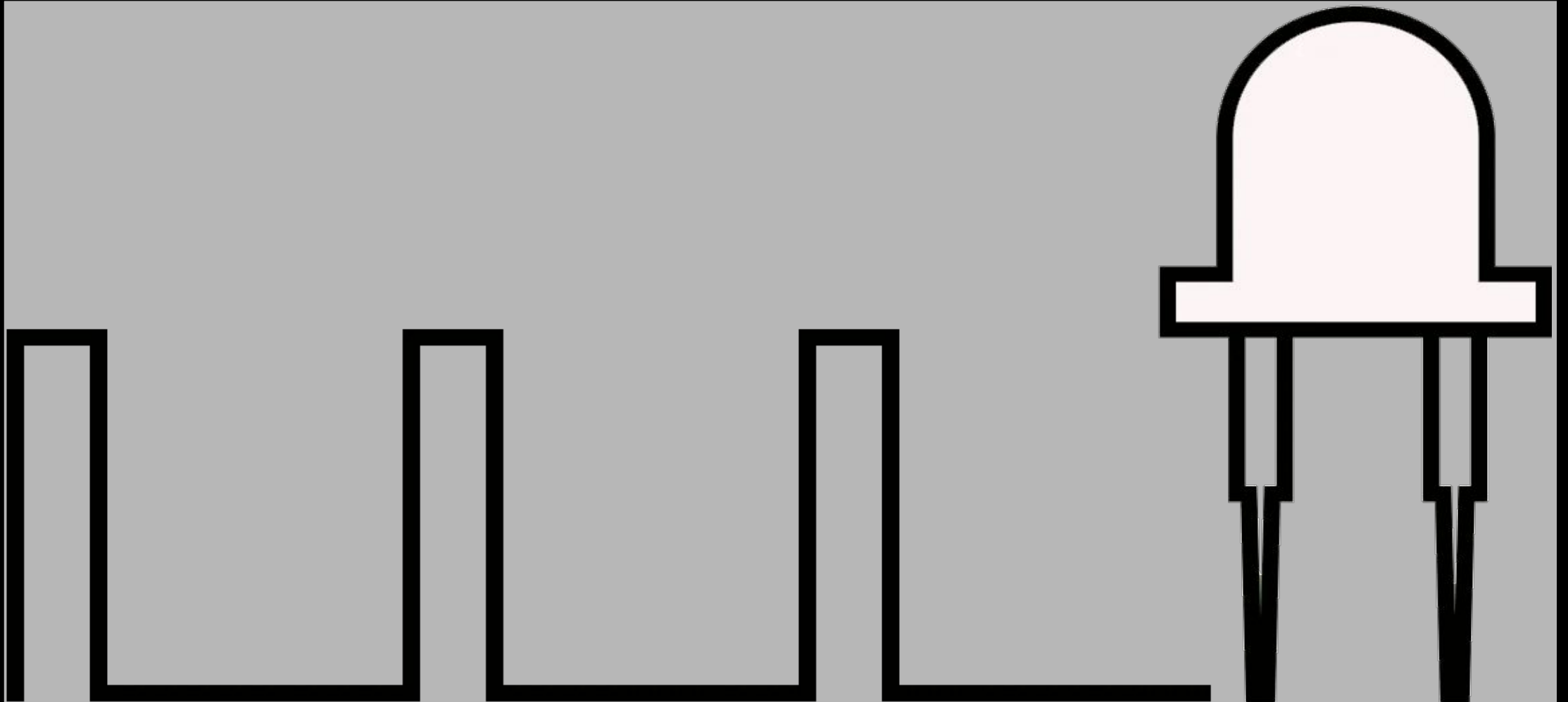
in- en uitfaden van LED, motor e.d.



# PWM - Pulse Width Modulation



# PWM - Pulse Width Modulation



# Al die andere dingen

**Welcome to Teensy® 4.0**  
32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click [Getting Started](http://www.pjrc.com/teensy).  
[www.pjrc.com/teensy](http://www.pjrc.com/teensy)

**Digital Pins**  
digitalRead  
digitalWrite  
pinMode

**Analog Pins**  
analogRead

**PWM Pins**  
analogWrite

**Digital Audio**  
Audio Library

**Serial Ports**  
Serial1 - Serial7

**I<sup>2</sup>C Port**  
Wire Library

**SPI Port**  
SPI Library

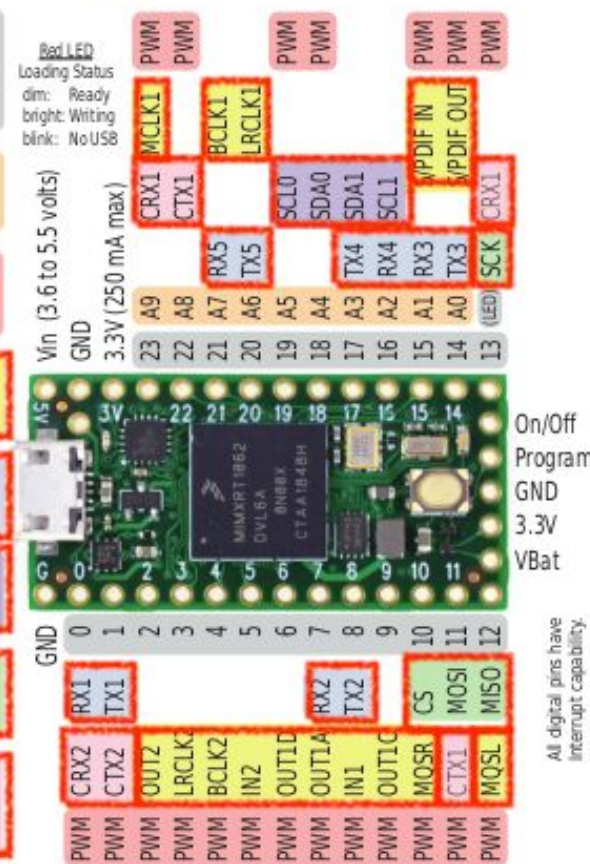
**CAN Bus**  
FlexCAN\_t4 Library

Red LED Loading Status  
dim: Ready  
bright: Writing  
blink: No USB

Min (3.6 to 5.5 volts)  
3.3V (250 mA max)

On/Off Program  
GND  
3.3V  
VBat

All digital pins have interrupt capability.



The diagram shows a Teensy 4.0 microcontroller board with its pins numbered 0 to 23. The pins are color-coded and labeled with their functions and associated libraries. The top row (pins 13-23) includes pins for digital I/O, PWM, and serial communication. The middle row (pins 0-12) includes pins for digital I/O, SPI, and CAN bus. The bottom row (pins 13-23) includes pins for digital I/O, PWM, and serial communication. The diagram also shows the location of the 3.3V and GND pins, and the location of the USB connector.

Verschillende protocollen voor:

- digitale audio
- seriële communicatie (bijv. een bluetooth module)
- complexere sensoren (GPS, accelerometer) of output (led-bars, displays)

# gedrag programmeren

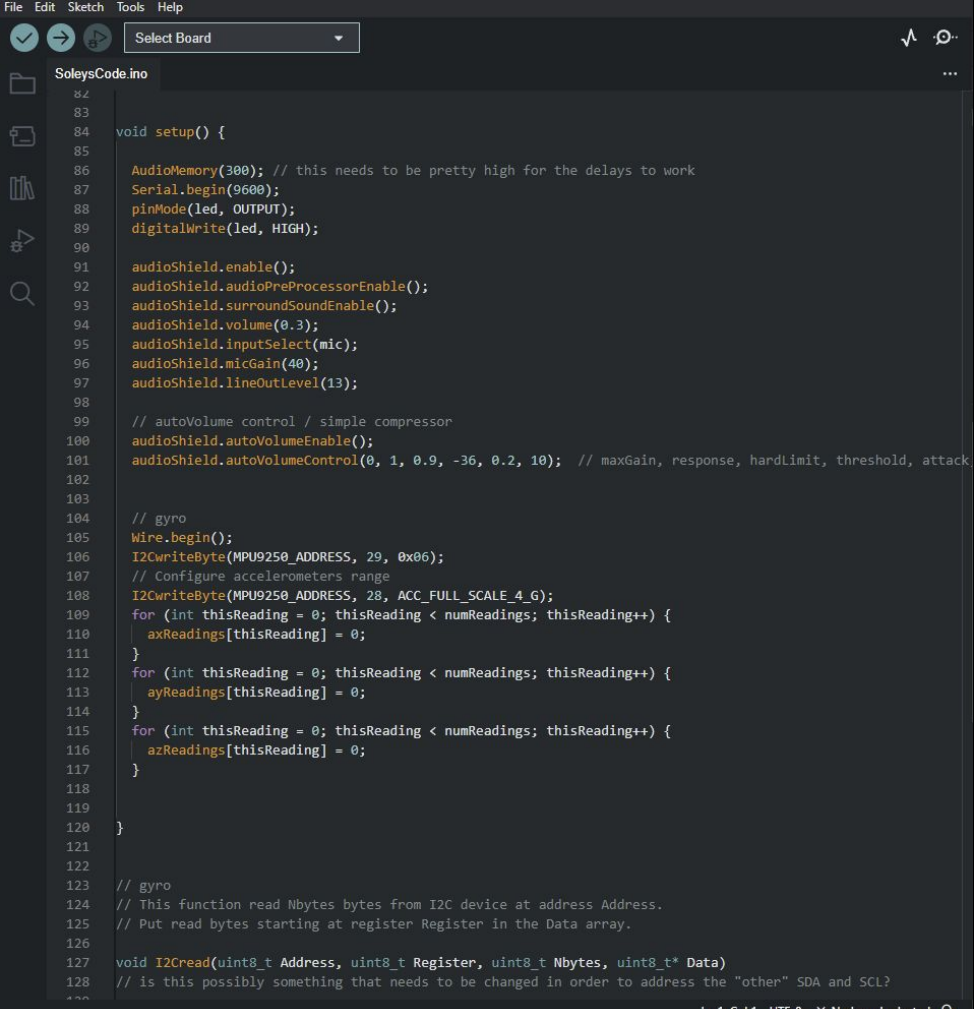
Jij bepaalt wat de Teensy doet  
met behulp van de Arduino-language  
gebaseerd op C

Lijkt in opzet veel op P5.js



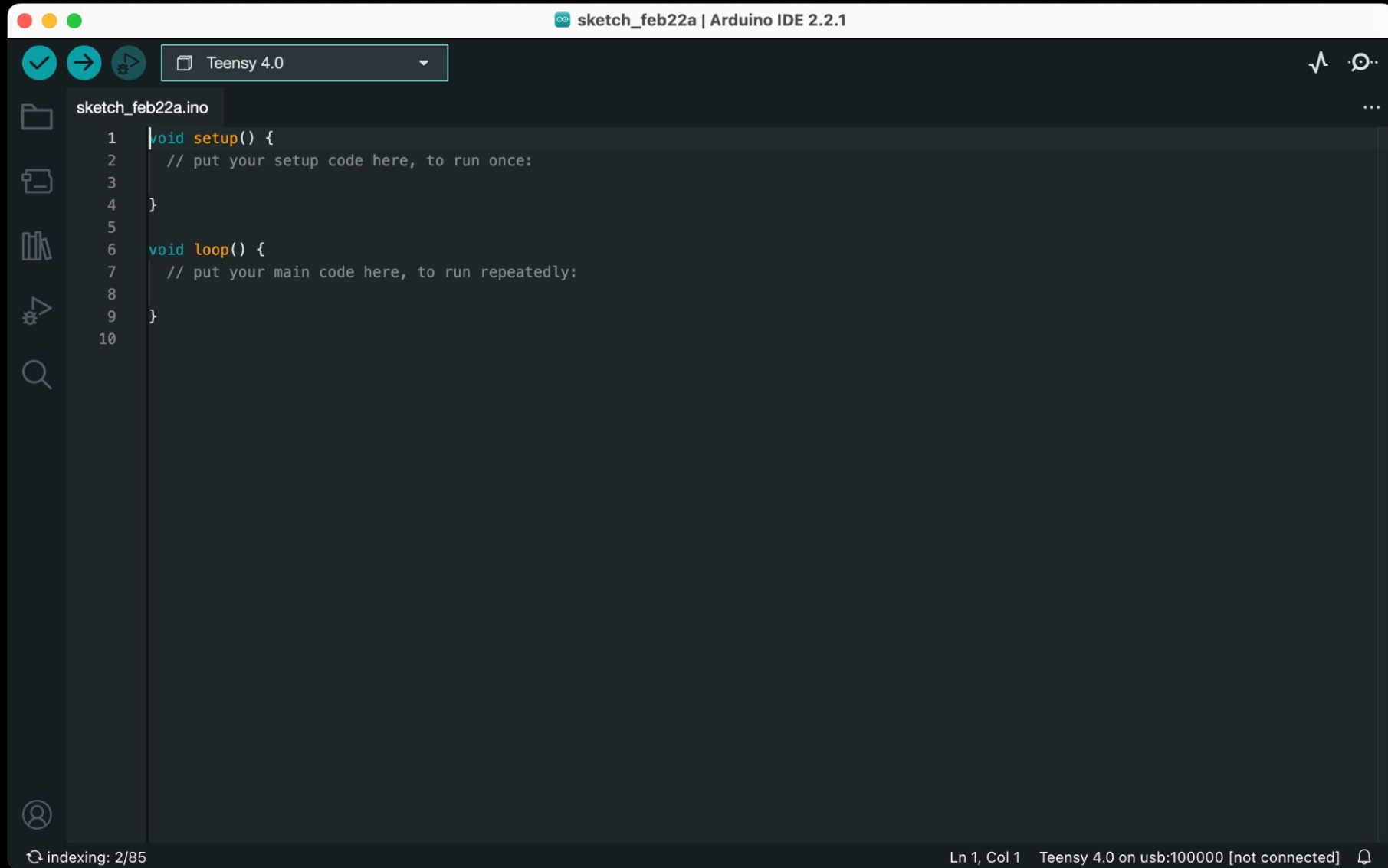
# Arduino IDE

- Software schrijven
- Software compileren
- Gecompileerde software uploaden
- Seriële communicatie met micro-controller



```
File Edit Sketch Tools Help
Select Board
SoleysCode.ino
82
83
84 void setup() {
85
86   AudioMemory(300); // this needs to be pretty high for the delays to work
87   Serial.begin(9600);
88   pinMode(led, OUTPUT);
89   digitalWrite(led, HIGH);
90
91   audioShield.enable();
92   audioShield.audioPreProcessorEnable();
93   audioShield.surroundSoundEnable();
94   audioShield.volume(0.3);
95   audioShield.inputSelect(mic);
96   audioShield.micGain(40);
97   audioShield.lineOutLevel(13);
98
99   // autoVolume control / simple compressor
100  audioShield.autoVolumeEnable();
101  audioShield.autoVolumeControl(0, 1, 0.9, -36, 0.2, 10); // maxGain, response, hardLimit, threshold, attack
102
103
104  // gyro
105  Wire.begin();
106  I2CwriteByte(MPU9250_ADDRESS, 29, 0x06);
107  // Configure accelerometers range
108  I2CwriteByte(MPU9250_ADDRESS, 28, ACC_FULL_SCALE_4_G);
109  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
110    axReadings[thisReading] = 0;
111  }
112  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
113    ayReadings[thisReading] = 0;
114  }
115  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
116    azReadings[thisReading] = 0;
117  }
118
119 }
120
121
122 // gyro
123 // This function read Nbytes bytes from I2C device at address Address.
124 // Put read bytes starting at register Register in the Data array.
125
126
127 void I2Cread(uint8_t Address, uint8_t Register, uint8_t Nbytes, uint8_t* Data)
128 // is this possibly something that needs to be changed in order to address the "other" SDA and SCL?
```

# Arduino IDE



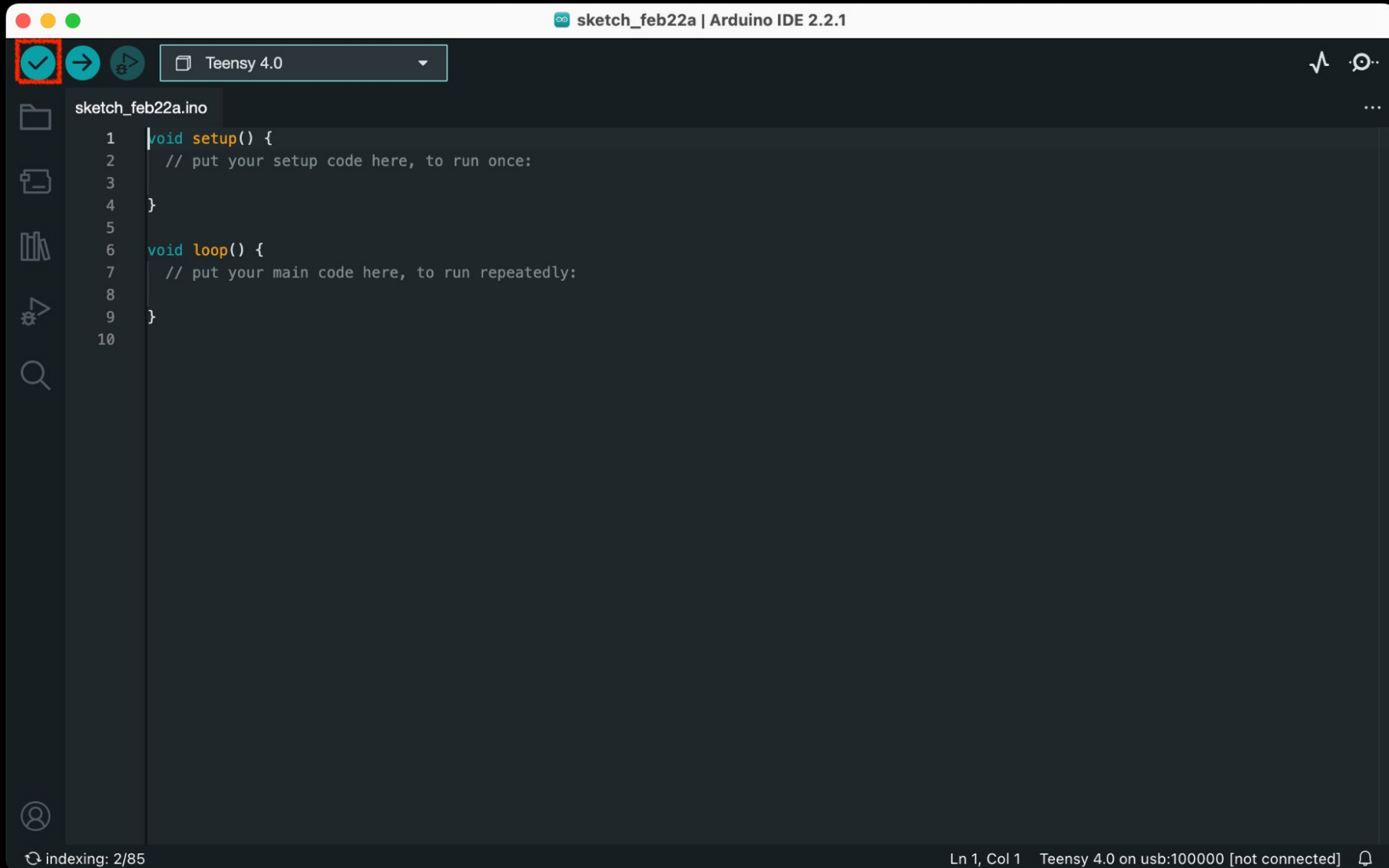
# Arduino IDE - code

```
sketch_feb22a.ino | Arduino IDE 2.2.1 | Teensy 4.0
```

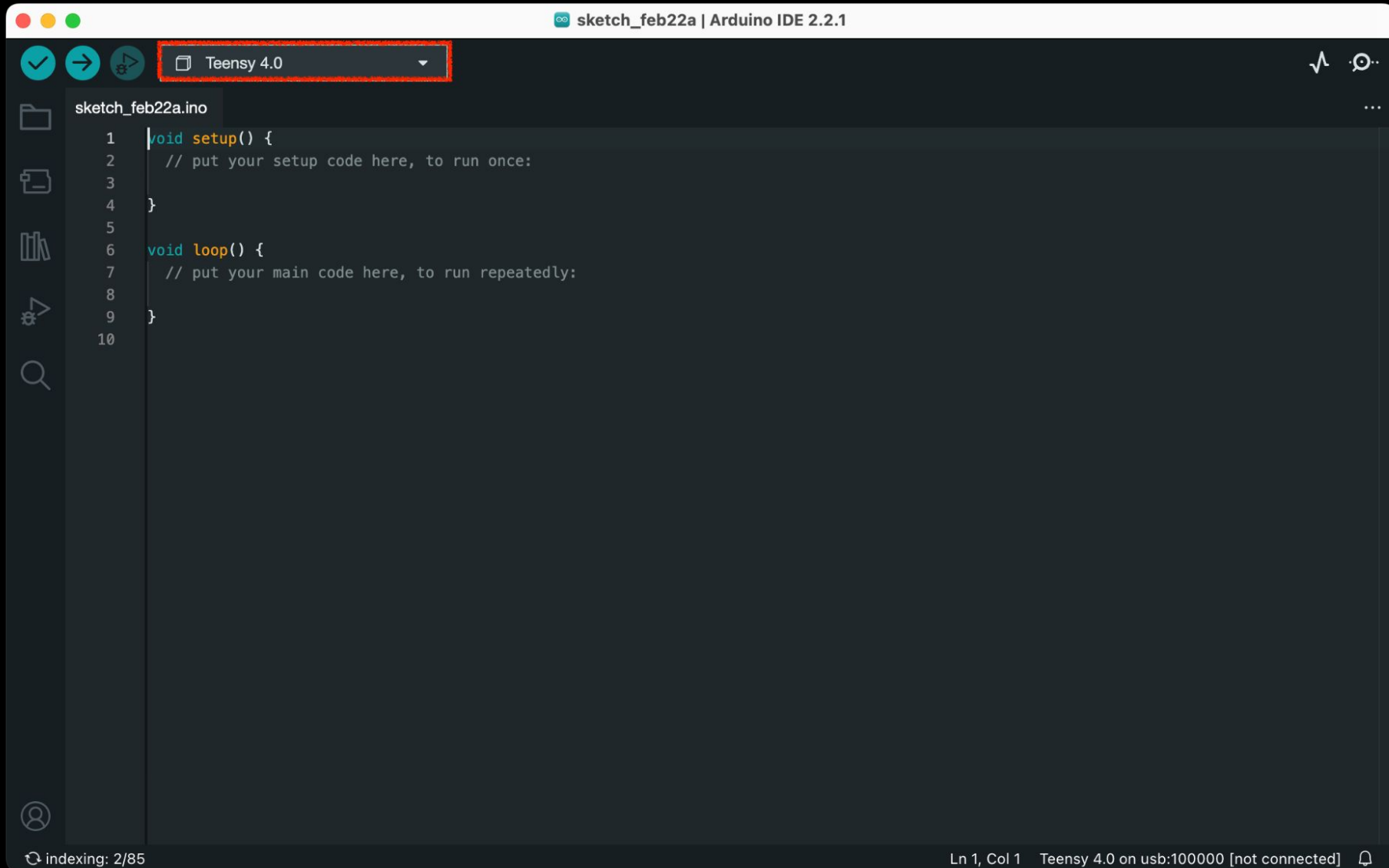
```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10
```

indexing: 2/85 | Ln 1, Col 1 | Teensy 4.0 on usb:100000 [not connected]

# Arduino IDE - check (cmd / ctrl) + r



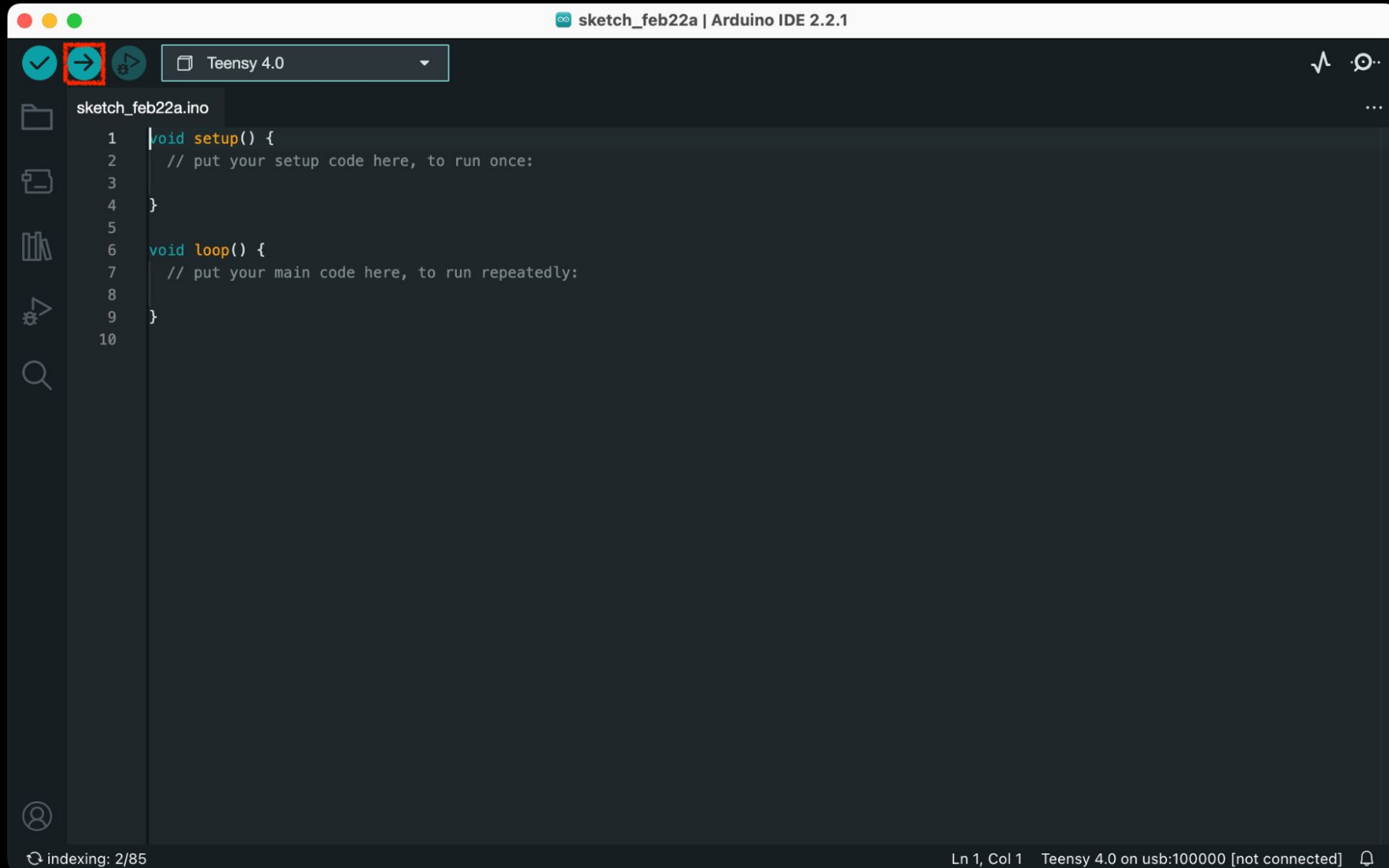
# Arduino IDE - select board





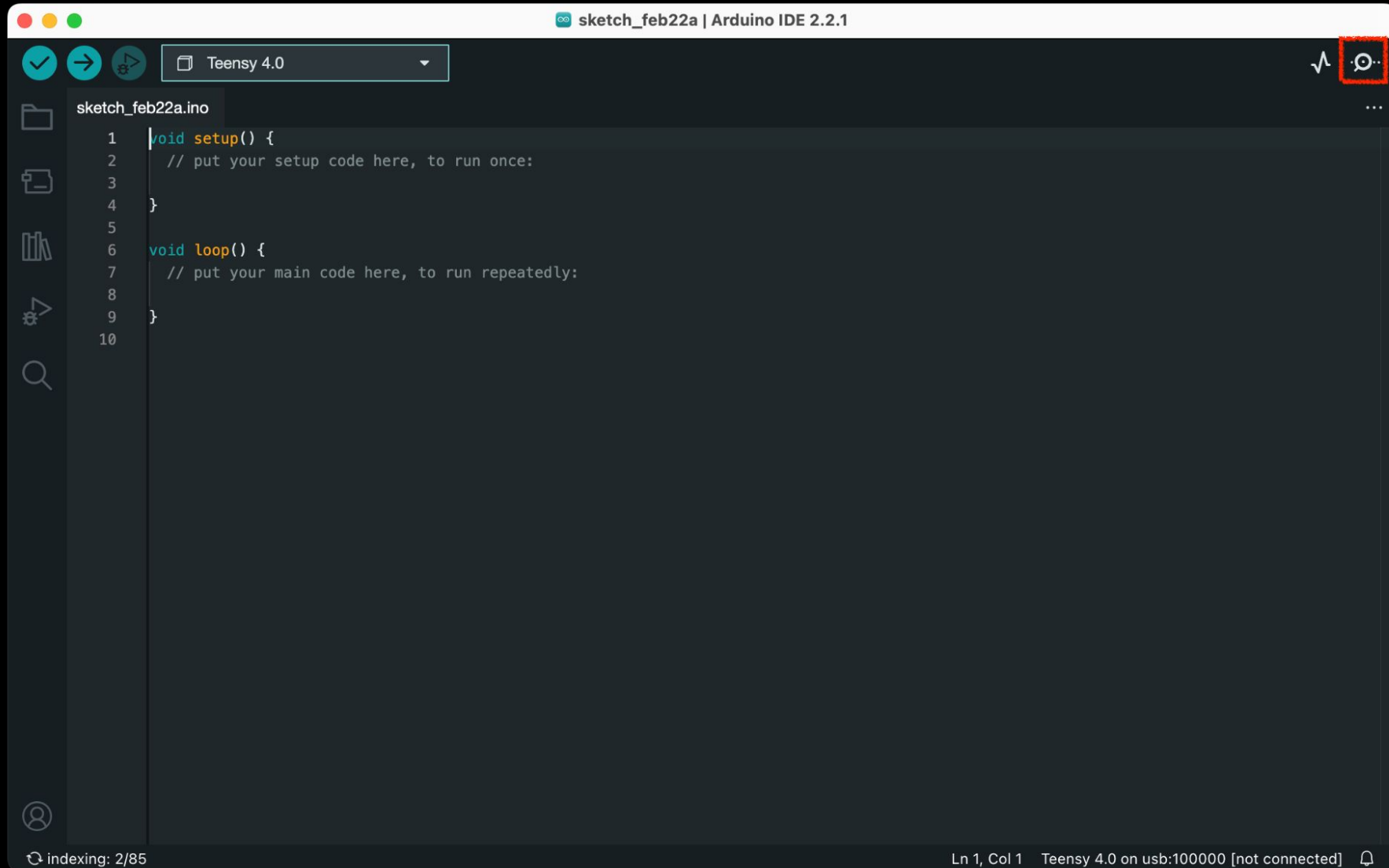
# Arduino IDE - upload

(cmd / ctrl) + u



# Arduino IDE - seriële monitor

(cmd / ctrl) + shift + m



# Teensy workflow



- Gecompileerde code vanuit Arduino IDE
- Seriële communicatie van en naar Teensy

# Arduino IDE

## Javascript:

- Draait in browser in Runtime
- ; wordt vergeven
- Datatype is fluide (altijd let)
- `console.log("hoi")`

## Arduino (C):

- Draait gecompileerd op micro-controller
- ; wordt niet vergeven
- Datatype staat vast (van te voren aangeven welk type)
- `Serial.begin(9600);`  
in `setup()` en later:  
`Serial.println("hoi");`



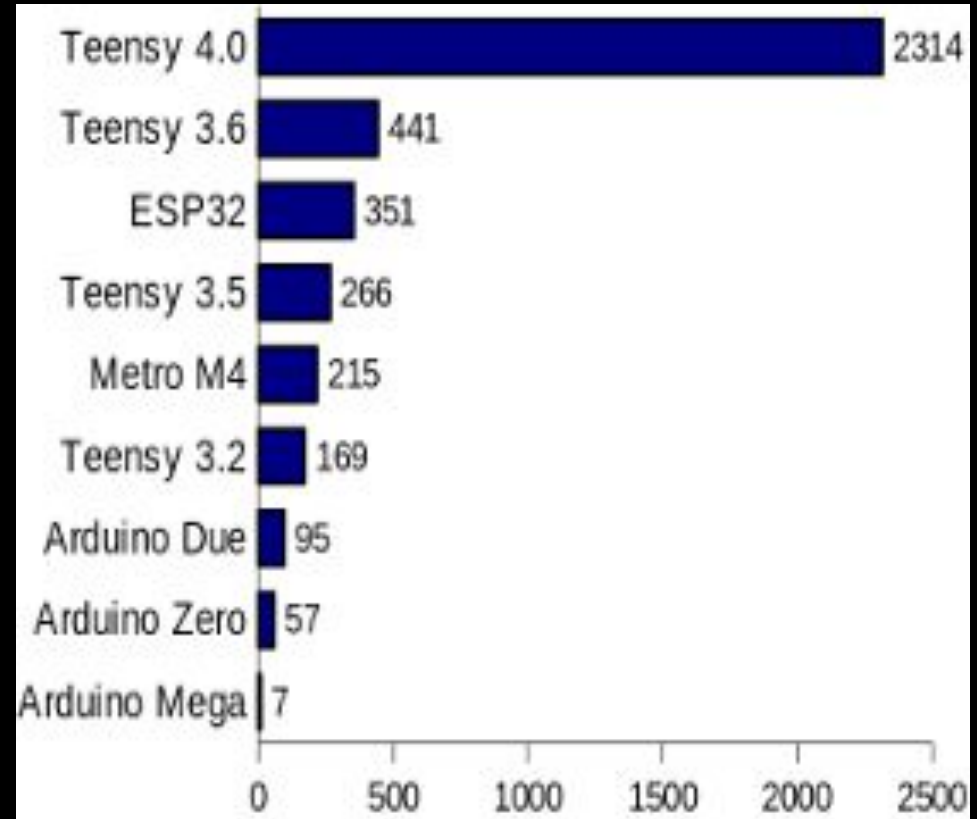
# Datatypes in Arduino

Type	Bereik	Voorbeeld declaratie
<b>int</b>	-32768 tot 32767	<code>int numLEDs = 5;</code>
<b>float</b>	-3.4028235E+38 tot 3.4028235E+38	<code>float maxHumanBodyTemp = 37.2;</code>
<b>boolean</b>	true    false	<code>bool ledIsOn = false;</code>
<b>byte</b>	0 tot 255	<code>byte redChannel = 0;</code>

# Arduino IDE

In Javascript/p5  
function setup()  
function draw()

In Arduino C  
void setup()  
void loop()



# Arduino basisfuncties

functie	werking
<code>pinMode(pin, modus)</code>	Stelt de werking van een fysieke pin in op <code>INPUT</code> , <code>INPUT_PULLUP</code> of <code>OUTPUT</code>
<code>digitalWrite(pin, state)</code>	Zet spanning op een pin <code>LOW</code> of <code>HIGH</code> (0v of 3.3v (5v op Arduino))
<code>digitalRead(pin)</code>	Leest spanning op pin ( <code>HIGH</code> of <code>LOW</code> / 1 of 0)
<code>analogRead(pin)</code>	Leest voltage op pin (waarde tussen 0 en 1023 / 0v en 3.3v)
<code>analogWrite(pin, byte)</code>	Zet variabel voltage op pin d.m.v. PWM
<code>delay(ms)</code>	Zet gehele microcontroller op pauze voor x-aantal milliseconden

# Arduino installeren (1/2)

## Downloads



### Arduino IDE 2.0.3

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

#### SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits

**Windows** MSI installer

**Windows** ZIP file

**Linux** Appliance 64 bits (X86-64)

**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.14: "Mojave" or newer, 64 bits

**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)




# Teensyduino installeren (2/2)

## Arduino 2.0.x Software Development

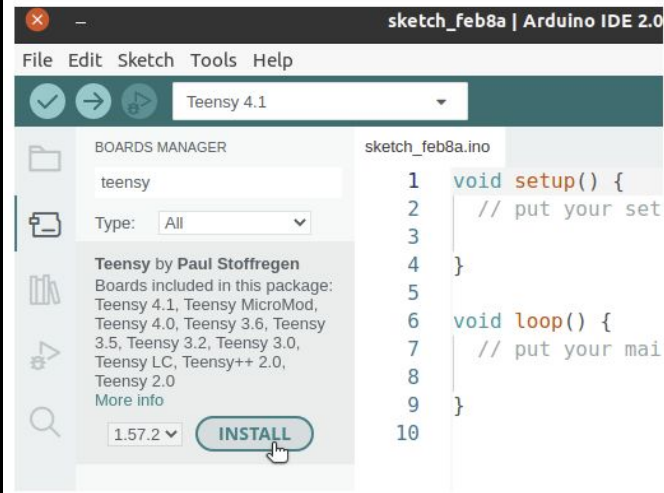
To install Teensy on Arduino IDE 2.0, click File > Preferences. In "Additional boards manager URLs", copy this link:

[https://www.pjrc.com/teensy/package\\_teensy\\_index.json](https://www.pjrc.com/teensy/package_teensy_index.json)

Verify code after upload  
 Auto save  
 Editor Quick Suggestions

Additional boards manager URLs:  

In the main Arduino window, open Boards Manager by clicking the left-side board icon, search for "teensy", and click "Install".



# Arduino IDE Seriële communicatie

In void setup()

```
Serial.begin(9600);
```

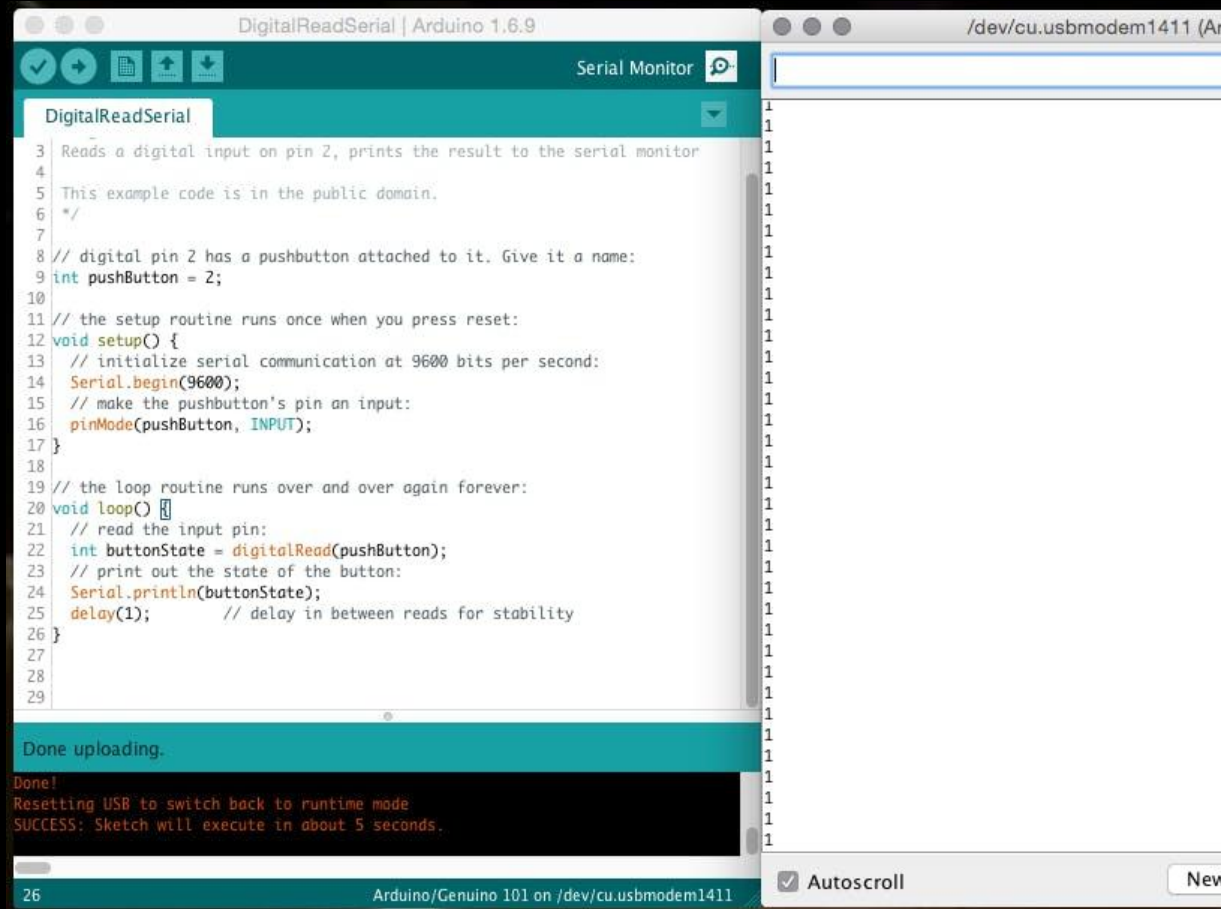
Daarna in bijv. loop()

```
Serial.println("hoi");
```

```
Serial.println(analogRead(4));
```

```
Serial.println("Value of potmeter on pin " + String(potPin) + " is now " + String(analogRead(potPin)));
```

# Arduino IDE Seriële monitor

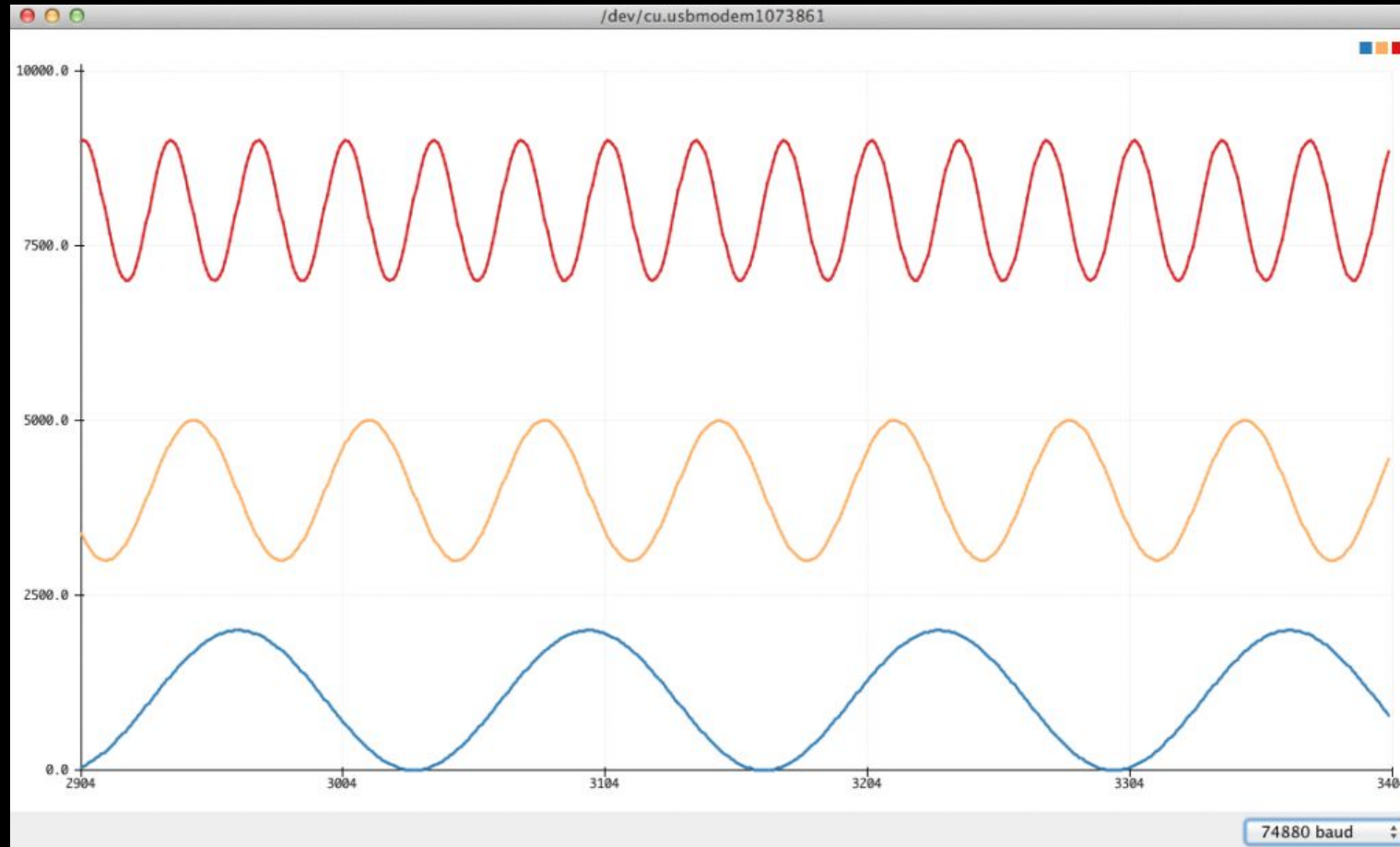


The image shows a screenshot of the Arduino IDE interface. The main window displays a sketch titled "DigitalReadSerial" with the following code:

```
3 Reads a digital input on pin 2, prints the result to the serial monitor
4
5 This example code is in the public domain.
6 */
7
8 // digital pin 2 has a pushbutton attached to it. Give it a name:
9 int pushButton = 2;
10
11 // the setup routine runs once when you press reset:
12 void setup() {
13   // initialize serial communication at 9600 bits per second:
14   Serial.begin(9600);
15   // make the pushbutton's pin an input:
16   pinMode(pushButton, INPUT);
17 }
18
19 // the loop routine runs over and over again forever:
20 void loop() {
21   // read the input pin:
22   int buttonState = digitalRead(pushButton);
23   // print out the state of the button:
24   Serial.println(buttonState);
25   delay(1);      // delay in between reads for stability
26 }
27
28
29
```

Below the code editor, the Serial Monitor window is open, showing the output of the sketch. The output consists of a series of "1" characters, indicating that the button is pressed. The Serial Monitor window also shows the status "Done uploading." and "SUCCESS: Sketch will execute in about 5 seconds." at the bottom.

# Arduino IDE Seriële plotter





Demonstratie