

Arduino Programming

Theorie Sessie 1



Introductie



ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

Wat is Arduino ?

Oorsprong

Arduino is ontstaan in 2005 in het
“Interactive Design Institute”
van Ivrea, Italië

“Massimo Banzì” is de bekendste figuur
en een van de oprichters van het Arduino project.



Wat is Arduino ?

Oorsprong

Massimo realiseerde zich dat microcontrollers een belangrijke element was om Interactieve Designs te creëren...

.... echter dat microcontrollers voor zijn studenten, veel eer artistieke dan technische mensen, héél erg ontoegankelijk was.



Wat is Arduino ?

Oorsprong

Alle microcontroller ontwikkel omgevingen van dat moment waren simpelweg té complex voor mensen met een beperkte technisch achtergrond

Alsof dat nog niet genoeg was waren ze hiernaast ook nog héél erg duur in aankoop,

Bovendien waren ze ook nog closed source en dus ontoegankelijk voor studenten om te bestuderen hoe ze werken...



Wat is Arduino ?

Oorsprong

Massimo besloot daarom om zélf een nieuwe ontwikkel omgeving te creëren met de volgende voorwaarden:

- héél toegankelijke ontwikkel omgeving
- eenvoudige “programmeertaal”
- eenvoudige programmer (zie later)
- low cost
- volledig opensource zowel soft- als hardware

Wat is Arduino ?

Oorsprong

Op zoek naar een naam voor het project besloten ze de naam “Arduino” te kiezen, dit naar “*Bar Di Re Arduino*” de naam van het café waar ze s’middags hun koffie gingen drinken :-)



Wat is Arduino ?

Onderdelen v/h ecosysteem

Het Arduino platform bestaat uit enkele onderdelen:

- de Arduino ontwikkel omgeving
- de Arduino programmer
- de Arduino hardware

Wat is Arduino ?

Integrated Development Environment

De ontwikkel omgeving
en een *stukje* van programmer
zijn bij “Arduino” geïntegreerd
in één software pakket, nl:

De Arduino IDE



Wat is Arduino ?

Hardware

Ook de hardware is reeds een integratie van enkele elementen, namelijk:

- Een microcontroller
- De bootloader
- USB controller
- Voedings spannings regeling

Arduino Microcontroller

Wat is een microcontroller?

een microcontroller is:
“een computer in één chip”



Arduino Microcontroller

Wat is een microcontroller?

In zo'n microcontroller chip vinden we minimaal:

- CPU
- Geheugen
- I/O systeem
- Seriële Communicatie



Arduino Microcontroller

Wat is een microcontroller?

Zo'n microcontroller heeft een soort
“ingebakken programmertaal”

Dit zijn een reeks instructies
die geprogrammeerd kunnen worden
door eentjes en nullen te schrijven
in het geheugen van de microcontroller.

Arduino Microcontroller

Wat is een microcontroller?

Deze instructies kunnen slechts kleinere 'getallen' verwerken dan een desktop of laptop CPU.

PC = meestal 64-bit instructies

In een microcontroller gaat het om 8-bits, 16-bits, of bij de hele krachtige om 32-bits instructies



Arduino Microcontroller

Wat is een microcontroller?

Afhankelijk van het type maken Arduino's gebruik van één de volgende microcontroller chips:

ATMega168 (8), ATMega328 (8),
ATmega32u4 (8), ATmega2560 (8),
Atheros AR9331 (16),
AT91SAM3X8E (32), ATSAMD21G18 (32)



Arduino Microcontroller

Wat is een microcontroller?

Een microcontroller heeft echter in principe géén:

- input voor een Keyboard
- output voor een Monitor
- aansluiting voor Harddisk
- evenmin interne software om deze hardware aan te sturen

Arduino Microcontroller

Als al deze zaken afwezig zijn,
hoe kunnen we dan zo'n chip programmeren... ?



Arduino Programmer

Om zo'n microcontroller te programmeren hebben een programmer nodig.



Arduino Programmer

Wat is een programmer ?

Dit is een toestel die:

- enerzijds kan communiceren met de chip
- anderzijds kan communiceren met een PC
- soft/hardware bevat die de chip in zijn programmeer modus kan plaatsen en data in de chip kan schrijven



Arduino Programmer

Wat is een programmer ?

Het hele idee is dat:

- We software ontwikkelen op de PC in een hogere programmeertaal
- De PC onze code zal vertalen (compileren) naar de instructies van de microcontroller
- We tenslotte via de programmer deze vertaalde code gaan uploaden in de chip



Arduino Programmer

Wat is een programmer ?

De slimme truuk van
de mensen van “Arduino” was:

- De hardware van zo’n programmer ineens in te bouwen op één pcb samen met de microcontroller.
- Software om de microcontroller te programmeren in diezelfde microcontroller in te bakken



Arduino Hardware

Types

Het verhaal van Arduino is gestart met de “Uno”

Deze “Uno” maakt gebruik van de ATmega328p microcontroller en is programmeerbaar via, en krijgt tevens zijn voedingsspanning via USB

Ondertussen zijn er ook een hele reeks andere Arduino's op de markt gekomen

Arduino Hardware

Entry Level



ARDUINO UNO



ARDUINO LEONARDO



ARDUINO 101



ARDUINO ESPLORA



ARDUINO MICRO



ARDUINO NANO

Arduino Hardware

Enhanced Features

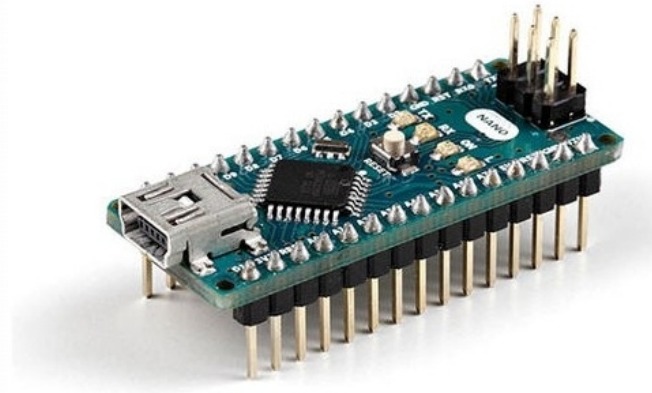
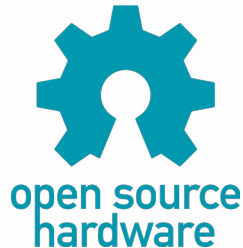


Arduino Hardware

Types

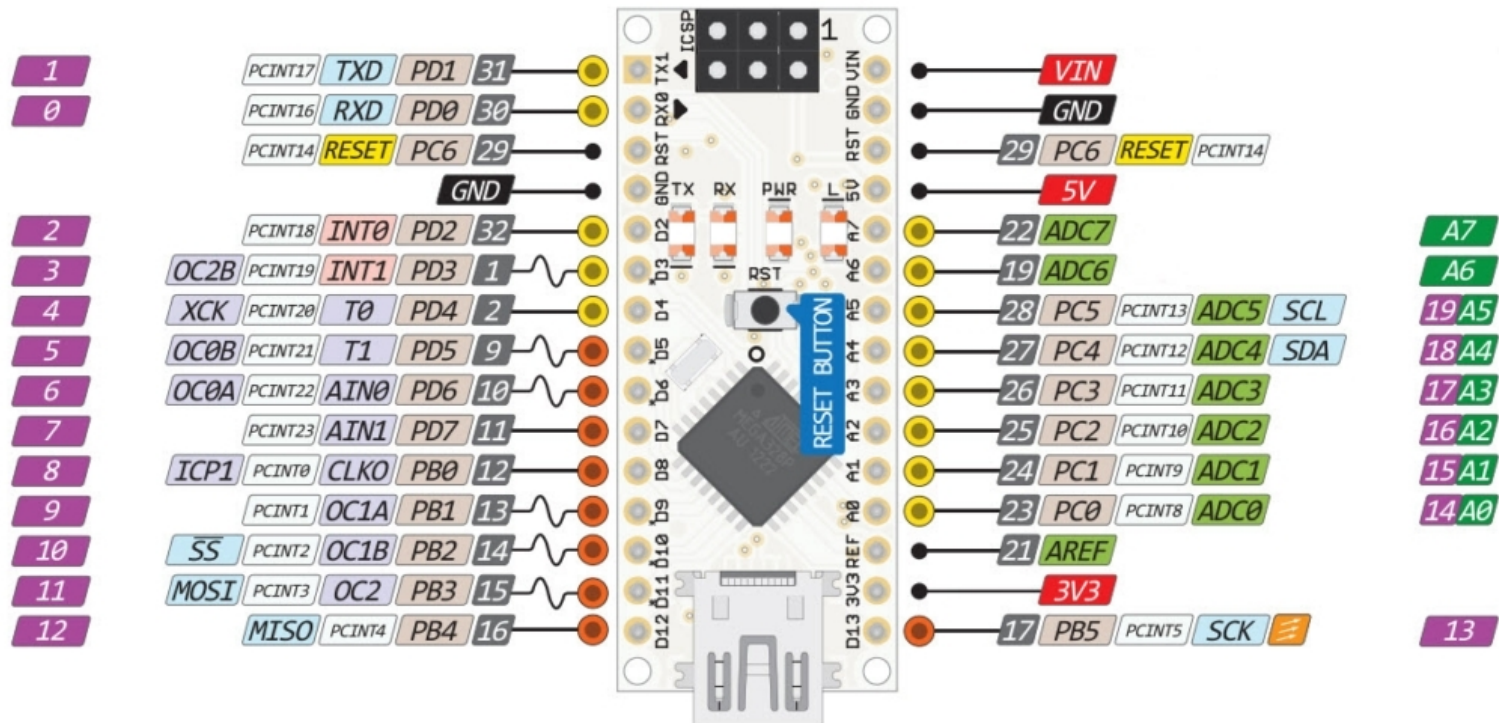
Voor het vak Arduino Programming
maken we gebruik van een (originele)

Arduino Nano



Arduino Hardware

Pin layout



Pin layout



Arduino IDE



Arduino IDE

Installatie

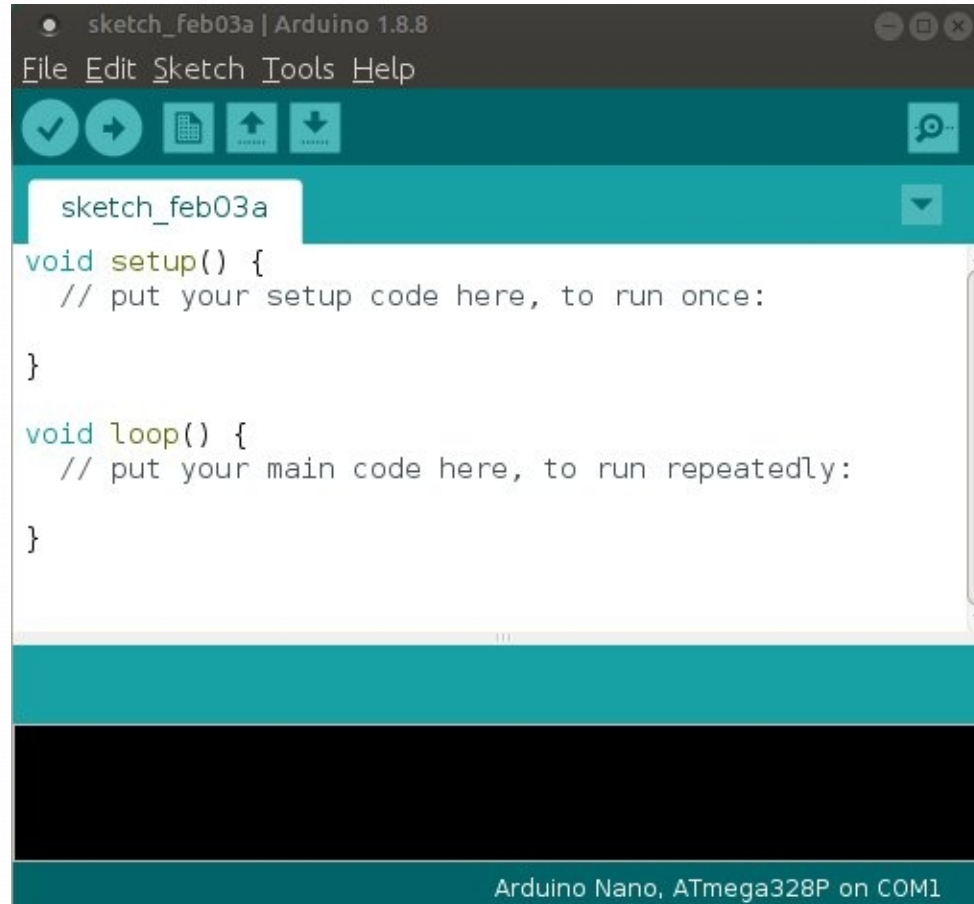
De Arduino IDE kan men downloaden
van de Arduino website

<https://www.arduino.cc/en/Main/Software>

De actuele versie is 1.8.8

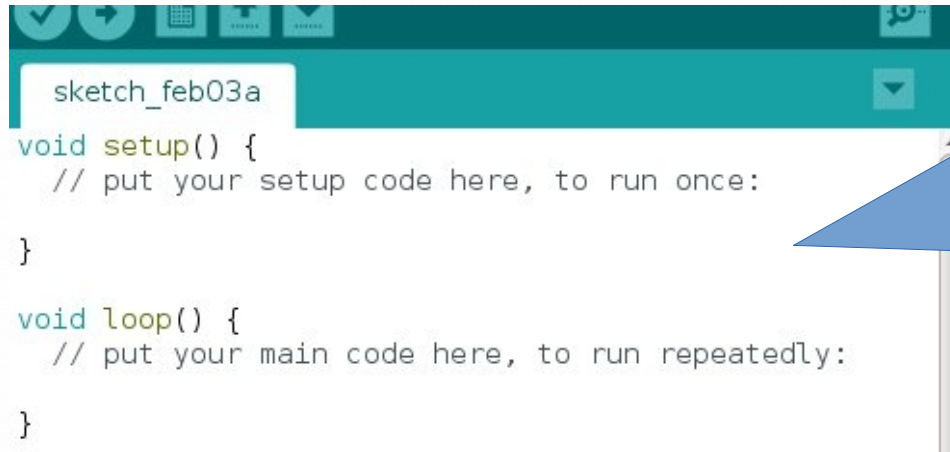


Arduino IDE



Arduino IDE

layout



```
sketch_feb03a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

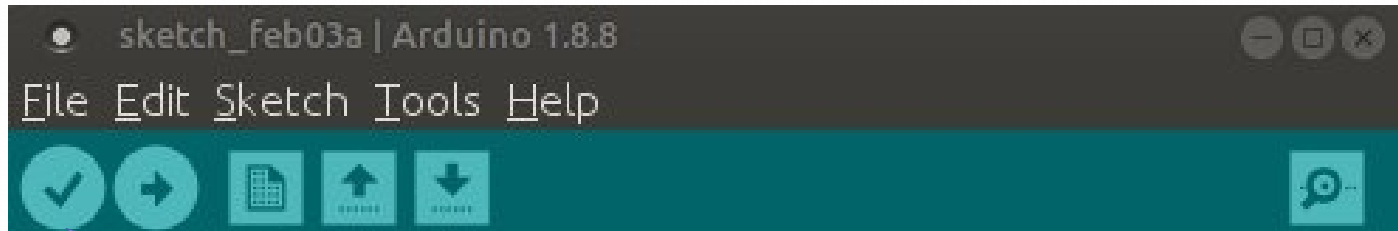
“Arduino Code”

hogere programmeertaal
code gebruikt om “sketches”
of arduino programma’s
in te schrijven

“Arduino Code” is gebaseerd
op de C/C++ familie
uitgebreid met een set
typische arduino commando’s

Arduino IDE

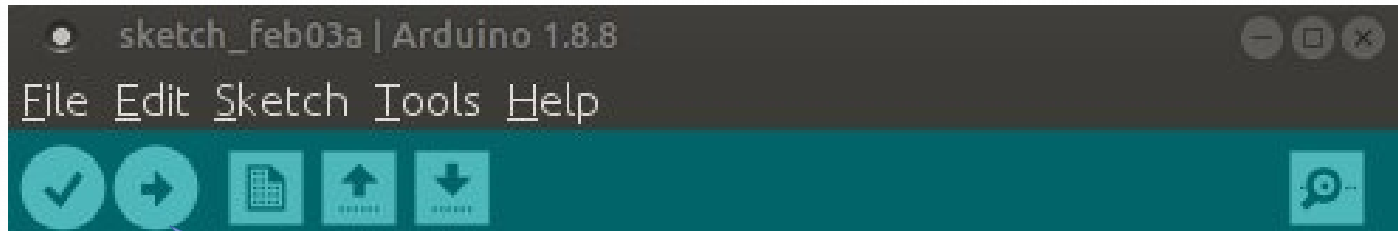
layout



Vertaalt de “Arduino-code” naar
de instructies van de microcontroller

Arduino IDE

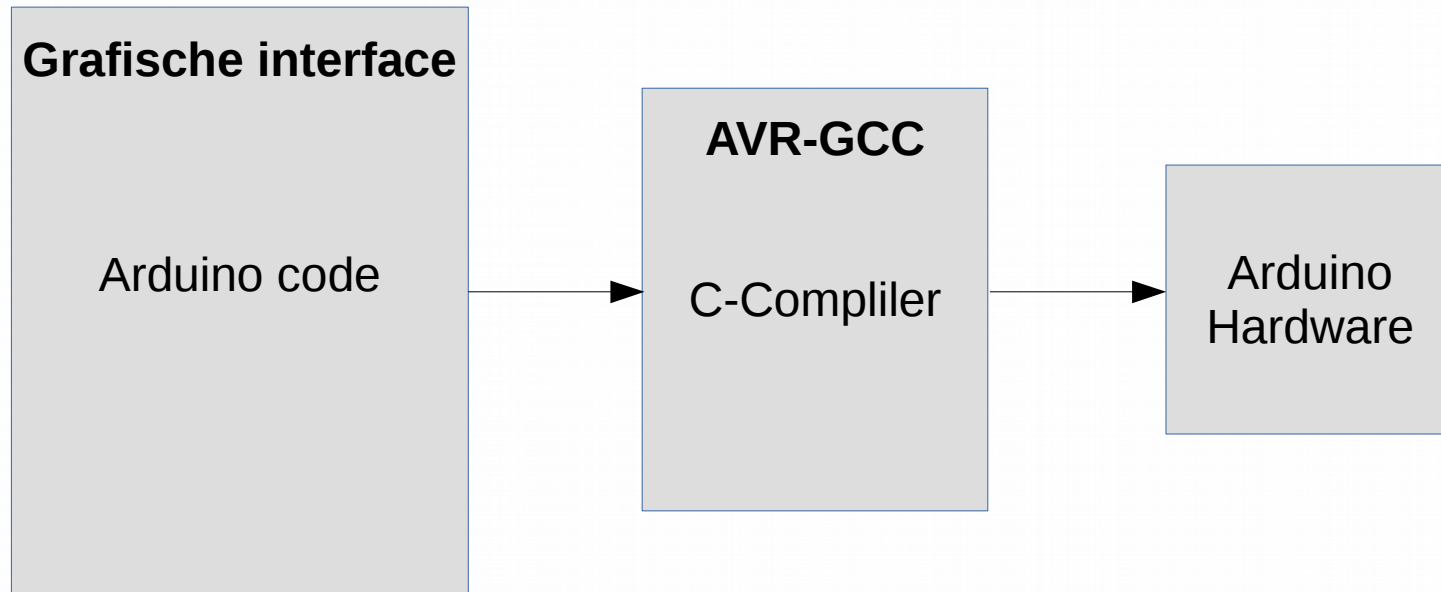
layout



Laad de vertaalde code
in het geheugen van de microcontroller chip
op de Arduino hardware

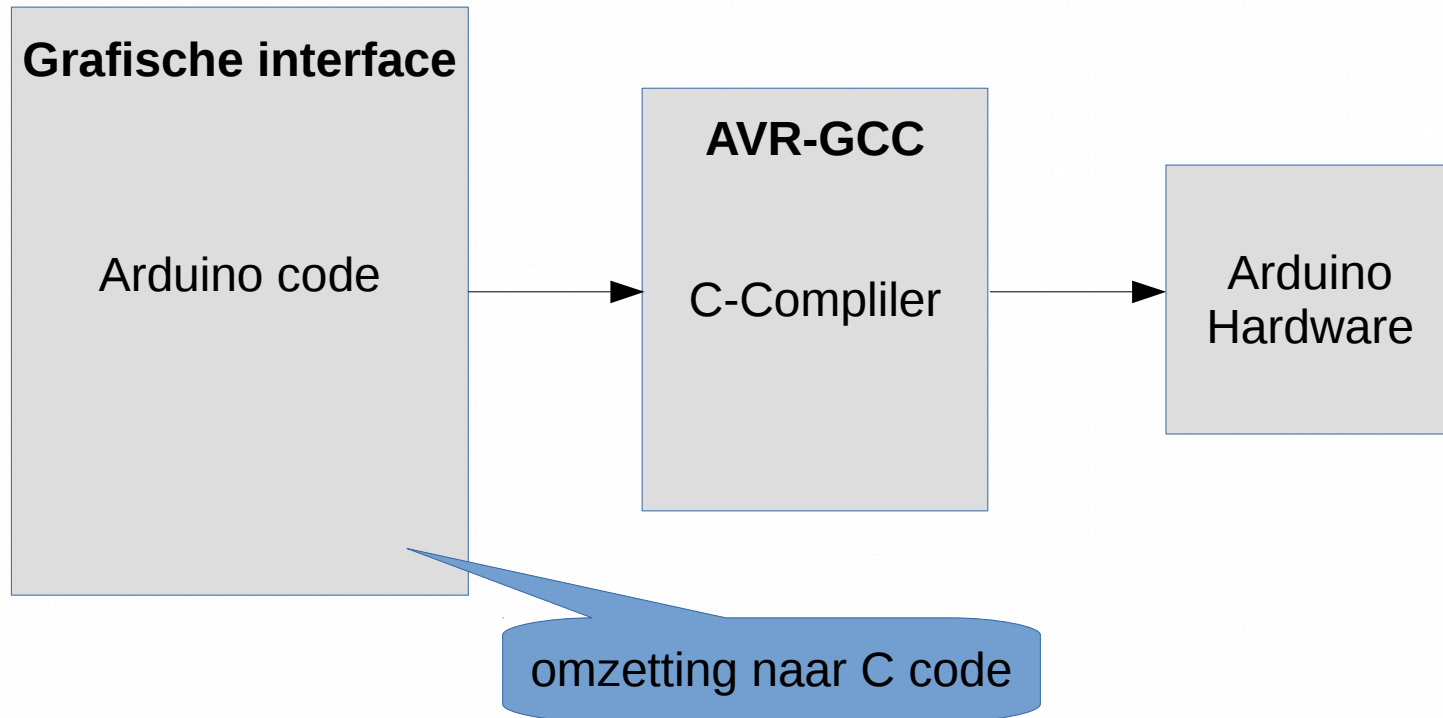
Arduino IDE

Interne opbouw



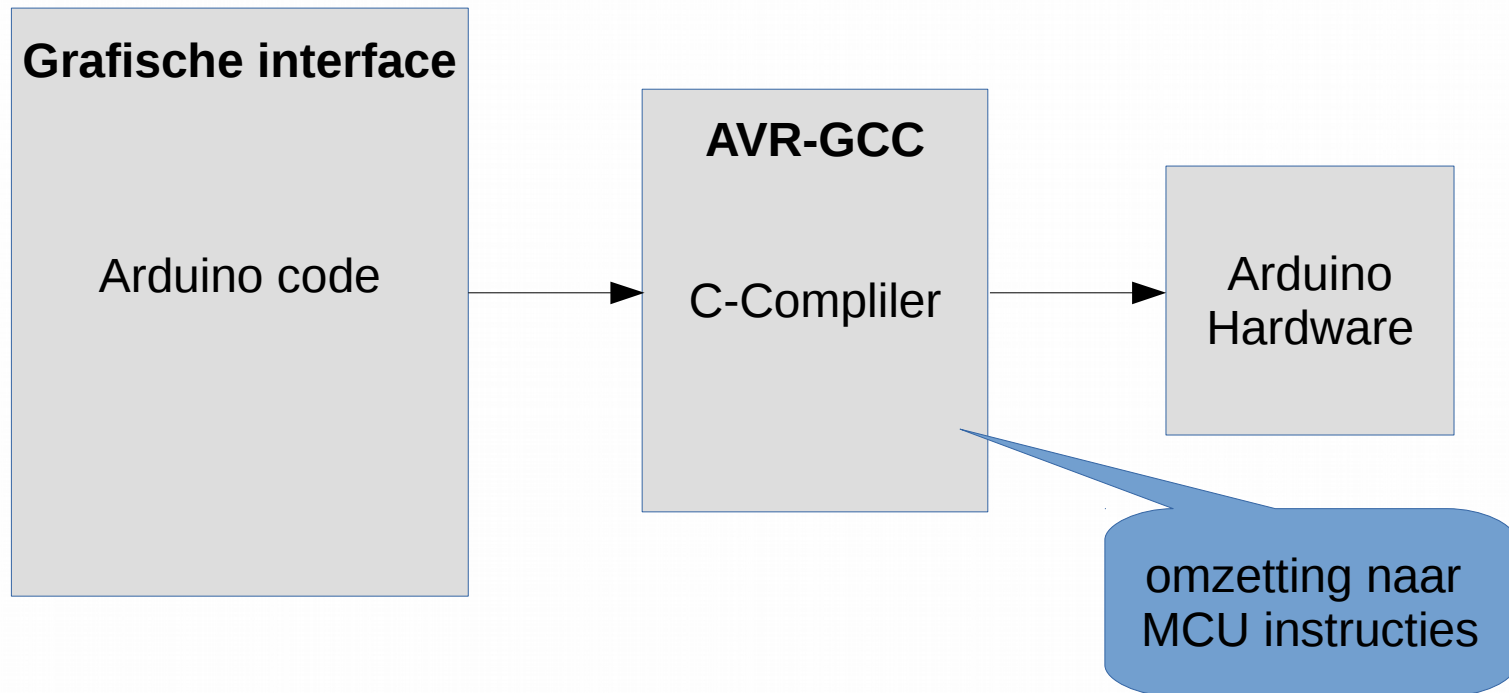
Arduino IDE

Interne opbouw



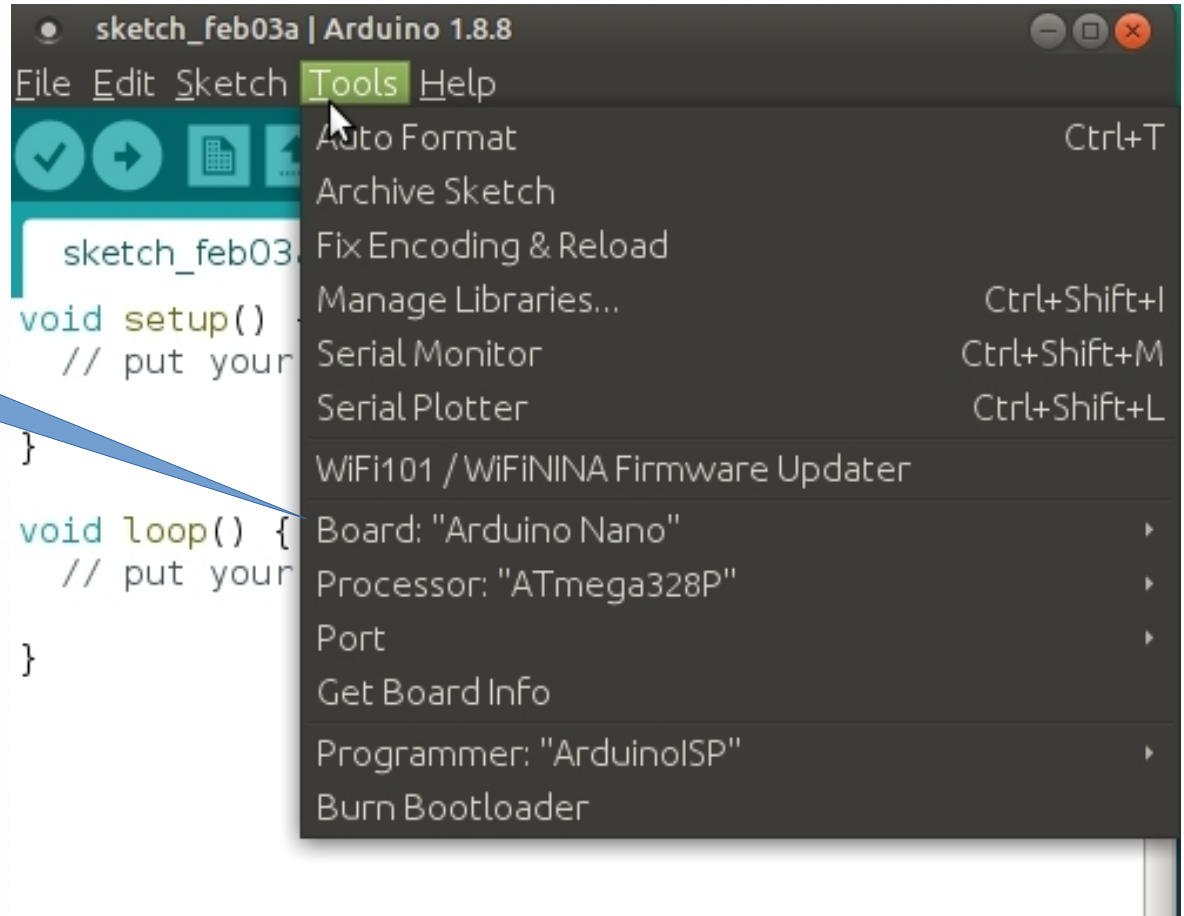
Arduino IDE

Interne opbouw



Arduino IDE

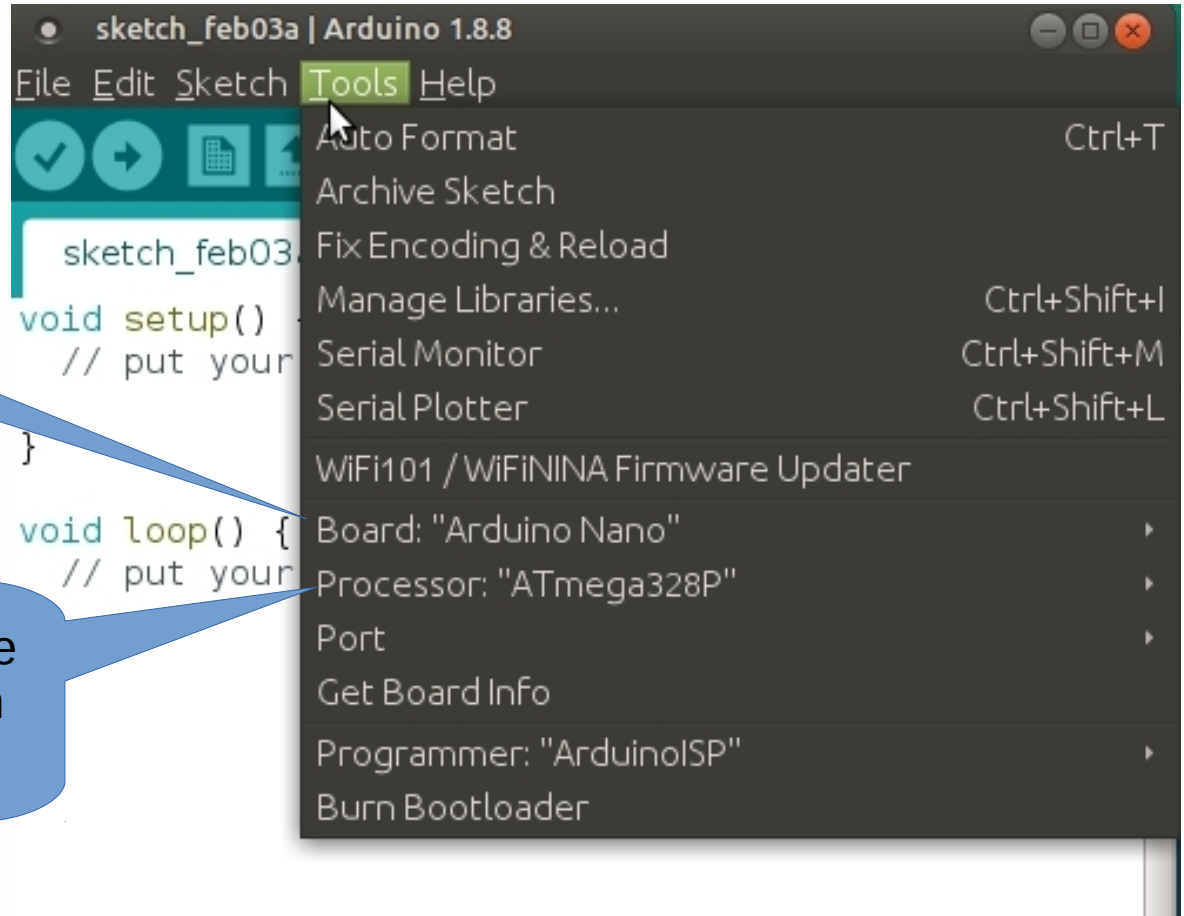
Keuze van de
gebruikte hardware



Arduino IDE

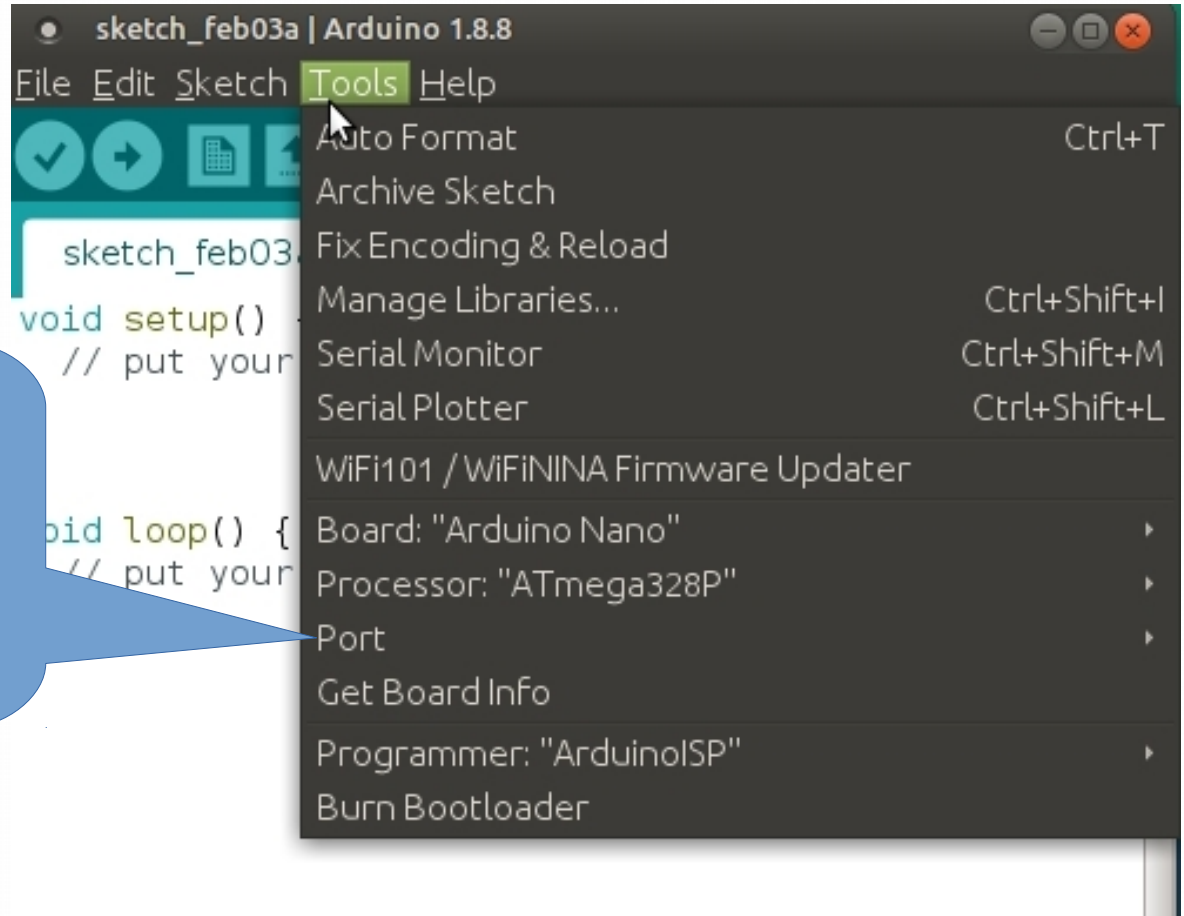
Keuze van de
gebruikte hardware

Indien verschillende
versies, keuze van
de gebruikte MCU



Arduino IDE

USB poort die
gebruikt wordt
om met de
Arduino Hardware
te communiceren



Arduino Code



Arduino Code

Code opdeling

A screenshot of the Arduino IDE interface. The top toolbar contains icons for checking, running, saving, and uploading code. The file explorer on the left shows a sketch named 'sketch_feb03a'. The main editor area displays C++ code with comments in Dutch explaining the structure of the code.

```
sketch_feb03a §  
  
// initialisatie gedeelte  
  
void setup() {  
    // configuratie gedeelte  
}  
  
void loop() {  
    // lus gedeelte  
  
    // meestal hoofdbrok van de code  
}
```

Arduino Code

GPIO commando's

Configureren van de pin-richting :

```
void setup() {  
  pinMode ( 2, OUTPUT );  
  pinMode ( 3, INPUT );  
}
```

functie

pin nummer

D2

D3



Arduino Code

GPIO commando's

Besturen van de uitgangen:

```
void loop() {  
  digitalWrite (| 7, HIGH);  
  digitalWrite (| 8, LOW);  
}
```

status

pin nummer

D7

D8



Arduino Code

GPIO commando's

Lezen van de ingangen:

```
void loop() {  
    bool pin3 = digitalRead ( 3 );  
    bool pin4 = digitalRead ( 4 );  
}
```

status

pin nummer

D3

D4



Arduino Code

Variabelen

Boolean: “1” of “0”

```
void loop() {  
    bool pin3 = digitalRead ( 3 );  
    bool pin4 = digitalRead ( 4 );  
}
```

variable type



Arduino Code

Types variabelen (nano)

bool: *boolean*

“0” of “1”, “True” of “False”, “High” of “Low”

byte: *byte* (8 bits getal)

getal van 0 tot 255 decimaal

getal van 00000000 tot 11111111 binair

getal van 00 tot FF hexadecimaal



Arduino Code

Types variabelen (nano)

int:

integer (2's complement 16 bits getal)

getal van -32768 tot 32767 decimaal

getal van 0000 tot FFFF hexadecimaal

unsigned int:
word:

integer zonder teken (16 bits getal)

getal van 0 to 65535 decimaal

getal van 0000 tot FFFF hexadecimaal



Arduino Code

Types variabelen (nano)

long:

lange integer (2's complement 32 bits getal)

getal van -2147483648 tot 2147483647 dec.
getal van 00000000 tot FFFFFFFF hex.

unsigned long:

lange integer zonder teken (32 bits getal)

getal van 0 tot 4294967295 decimaal
getal van 00000000 tot FFFFFFFF hex.



Arduino Code

Types variabelen (nano)

float:

floating point (32 bits getal, wetenschappelijk notatie)

getal van 3.4028235 E+38
tot -3.4028235 E+38

double:

“grote” float (32 bits getal, wetenschappelijk notatie)

getal van 3.4028235 E+38
tot -3.4028235 E+38

is er dan géén verschil met float...?



Arduino Code

Types variabelen

Types zijn afhankelijk van de gebruikte microcontroller !

<u>int:</u>	= 16-bit's	→	Arduino Uno en Nano
	= 32-bit's	→	Arduino Duo en SAMD boards
<u>float:</u>	= 32-bit's	→	Op alle boards
<u>double:</u>	= 32-bit's	→	Arduino Uno en Nano
	= 64-bit's	→	Arduino Duo en SAMD boards

Arduino Code

Types variabelen

Types zijn afhankelijk van de gebruikte microcontroller !

Waarom... ?

Niet elke microcontroller kan alle bewerkingen uitvoeren.

De gebruikte 8-bits microcontrollers kunnen geen bewerkingen uitvoeren met 64-bits getallen.

Denk aan een gewone en een wetenschappelijke rekenmachine



Arduino Code

Types variabelen (nano)

char:

character (8-bits, één letter, ascii)

één letter uit de ASCII tabel

a-z, A-Z, 0-9, leestekens, speciale tekens

char []:

string (reeks van 8bits characters)

verzameling van tekens uit de ASCII tabel

De rij van characters wordt afgesloten met een “\0” of zogenaamde null-character



Arduino Code

Scope van variabelen

Net als in andere hogere talen
kunnen variabelen lokaal of globaal zijn :

```
int deze_variable_is_globaal;  
  
void setup() {  
    int deze_variable_is_lokaal;  
}  
  
void loop() {  
    int ook_deze_variable_is_lokaal;  
}
```

Arduino Code

Vreemde variabelen

Arduino pinnen:

Belangrijk is weten dat achterliggend de naam van de arduino-pinnen een getal waarde zit.

Arduino Code

Vreemde variabelen

Arduino pinnen:

Belangrijk is weten dat achterliggend de naam van de arduino-pinnen een getal waarde zit.

Volgende expressie's zijn bijgevolg volledig syntactisch correct:

```
int een_arduino_pin = 2;
```

```
int andere_arduino_pin = A3;
```

Arduino Code

Wachten

In elektronische schakelingen is een van de activiteiten die we vaak nodig hebben “*niets doen*” of wachten

Voorbeeld:

- Een LED laten branden voor 1 seconde
- En vervolgens doven voor 1 seconde

Omdat “wachten” zo vaak voorkomt heeft het Arduino team er speciaal enkele commando's voor gemaakt

Arduino Code

Wachten

```
delay(t);
```

dit commando zal gedurende “ t “
milliseconden niets doen.

```
t = millis();
```

dit commando retourneert het aantal
milliseconden die verstreken zijn sinds
power-on of reset van de hardware

```
t = micros();
```

dit commando retourneert het aantal
microseconden die verstreken zijn sinds
power-on of reset van de hardware

Arduino Code

“Hello world“

Hiermee hebben we nét genoeg
commando's gezien voor een éérste
“Arduino Sketch”

... en dat is een stukje code
die een LED doet knipperen
met een vast ritme,

ook wel bekend als de “blink-sketch”



Arduino Code

“Blink”

```
int ledpin = 3;

void setup() {

    pinMode ( ledpin, OUTPUT );

}

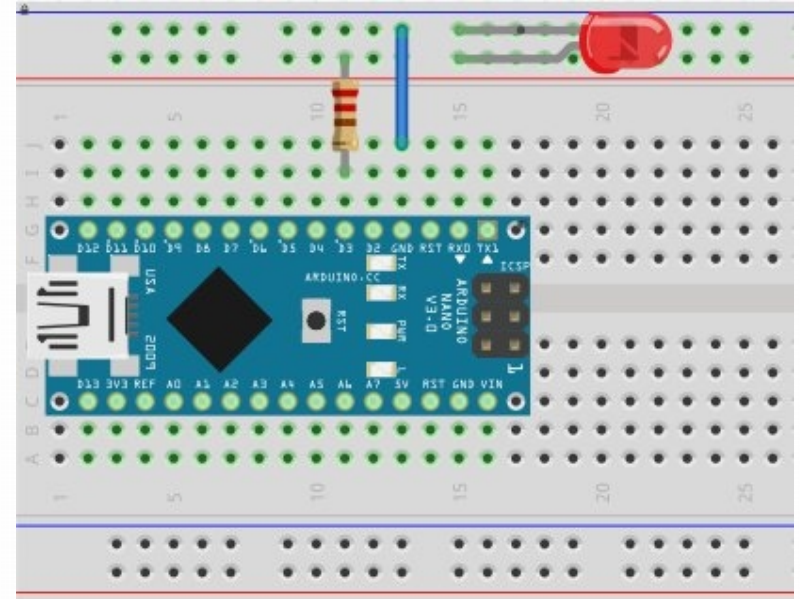
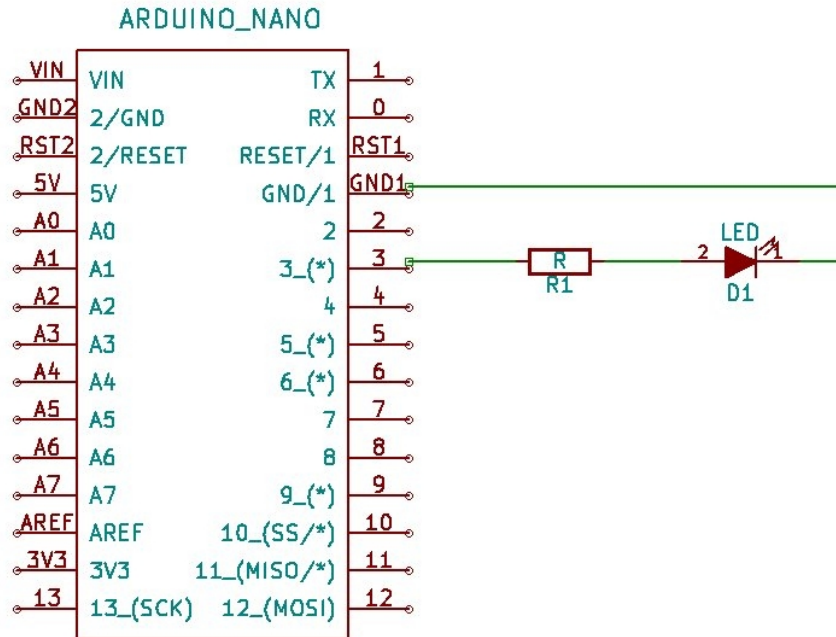
void loop() {

    digitalWrite ( ledpin, HIGH );
    delay(500);
    digitalWrite ( ledpin, LOW );
    delay(500);

}
```

Arduino Code

“Blink”



Arduino Code

Conditionele Statements

if .. else

```
if ( boolean_expression ) {  
    // deze code wordt uitgevoerd indien waar  
}  
else {  
    // deze code wordt uitgevoerd indien onwaar  
}
```

Arduino Code

```
int knop = 3;
int led = 4;

void setup() {
  pinMode ( knop, INPUT );
  pinMode ( led, OUTPUT );
}

void loop() {

  bool toestand = digitalRead(knop);

  if ( toestand ) {
    digitalWrite( led, HIGH );
  } else {
    digitalWrite( led, LOW );
  }
}
```


Arduino Code

Analog GPIO commando

Lezen van de analoge ingangen:

```
void loop() {  
  
  int analoog = analogRead(A3);  
  
}
```

analoge waarde
tussen 0 en 1023

analoge pin
nummer

A3



Arduino Code

Analog GPIO commando

Lezen analoge ingangen:

- Een spanning tussen 0 en 5V wordt een getal tussen 0 en 1023
- Dit wil zeggen dat elk stapje

$$5V / 1024 = 0,00488V = \underline{\underline{4,88 \text{ mV}}}$$

groot is.

A3

