

Leren programmeren, meten en sturen met de Arduino

Vierde druk, versie 2021

Jacco de Jong

Leeswijzer

ACTIE	stapsgewijs opzetten van hard- en software
SKETCH	programmaregels
FUNCTIES EN WAARDEN	uitleg van programmaonderdelen
OEFENING	creatief aan de slag
OEFENING*	oefeningen gemerkt met een * zijn voor differentiatie; overleg met je docent of je die moet maken

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: Jacco de Jong
Redactie en opmaak: Henk Pel
Titel: Leren programmeren, meten en sturen met de Arduino. Vierde druk, versie 2021
ISBN 978 90 372 6030 4
Eerste druk / eerste oplage
© Boom beroepsonderwijs 2021

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoud

	Inleiding—1
	Leerdoelen 1
1	Aftrap—3
1.1	Aanschaf 3
1.2	Arduino UNO R3 of Arduino MEGA 2560? 4
1.3	Poorten en voeding 5
1.4	Systeemeisen 6
1.5	Software 6
1.6	Arduino met pc verbinden 10
1.7	Software instellen 11
2	Eerste project—13
2.1	Knipperende LED 13
2.2	Sketch 15
2.3	Functies en waarden 17
2.4	Verifieer versus Upload 18
2.5	Oefeningen 19
	2.5.1 Oefening – Knippersnelheid aanpassen 19
	2.5.2 Oefening – Sketch opslaan en openen 19
	2.5.3 Oefening – Zonder USB-kabel 19
3	Werken met componenten—21
3.1	Breadboard 21
3.2	LED 22
3.3	Halfgeleider 23
3.4	Weerstand 24
3.5	LED en weerstand aansluiten 26
3.6	Oefeningen 27
	3.6.1 Oefening – Twee LEDs knippen 27
	3.6.2 Oefening – Versnelde LED 29
	3.6.3 Oefening – Externe LEDs zelfstandig laten knippen 29
	3.6.4 Oefening – Knight Rider 30
	3.6.5 Oefening* – Morsecode 30

4	Digitale input en output—31
4.1	Drukknopschakelaar 31
4.2	HIGH en LOW 32
4.3	In- en uitgangen (digitaal) 33
4.4	Pull-down-weerstand 34
4.5	Pull-up-weerstand 35
4.6	LED in- en uitschakelen 35
4.7	Digitale Hall-effect-sensor 38
	4.7.1 <i>Contactloze schakelaar</i> 39
	4.7.2 <i>AAN wordt UIT en UIT wordt AAN</i> 41
4.8	Oefeningen 42
	4.8.1 <i>Oefening – 30 seconden uitschakelvertraging</i> 42
	4.8.2 <i>Oefening – Looplicht van acht LEDs op schakelaar</i> 42
	4.8.3 <i>Oefening* – Veranderende functie van de schakelaar</i> 42
	4.8.4 <i>Oefening* – Toggle-schakelaar</i> 42
5	Analoge sensoren – deel 1—44
5.1	Spanningsdeler 44
5.2	Potentiometer 45
5.3	Regel knippersnelheid met potentiometer 46
5.4	Lichtsensoren LDR 47
5.5	LDR-spanningsdeler 49
5.6	Oefeningen 51
	5.6.1 <i>Oefening – Automatische verlichting</i> 51
	5.6.2 <i>Oefening* – Volkomen rood kruispunt</i> 52
6	Rekenen op Seriële monitor—54
6.1	Vermenigvuldigen 54
6.2	Optellen 57
6.3	Stopwatch 58
6.4	Oefeningen 59
	6.4.1 <i>Oefening – Aftellende klok</i> 59
7	Analoge sensoren – deel 2—60
7.1	Temperatuursensor LM35 60
7.2	Temperatuurmeting 62
7.3	Druksensor 65
7.4	Drukmeting 67
7.5	Gassensor MQ-x 70
7.6	Gasmeting (mg/l en ppm) 72

- 7.7 Oefeningen 73
 - 7.7.1 Oefening – Lumenmeter 73
 - 7.7.2 Oefening – Tellen 73
 - 7.7.3 Oefening* – Alcoholslot 73

- 8 Digitale sensoren—74**
 - 8.1 DHT11 temperatuur- en relatieve-luchtvochtigheidsensor 74
 - 8.2 Meten met de DHT11 75

- 9 Library—76**
 - 9.1 Wat is een library? 76
 - 9.2 Library installeren (standaard) 77
 - 9.3 Library toevoegen – include 78
 - 9.4 Library troubleshooting 79
 - 9.5 Library installeren (custom) 80
 - 9.6 Library gebruiken – vochtmeting met DHT11 83
 - 9.7 Samenvatting – werken met libraries 87
 - 9.8 Oefeningen 88
 - 9.8.1 Oefening – DHT11 tot op twee decimalen nauwkeurig 88
 - 9.8.2 Oefening* – Thermostaat en hygrometer 88

- 10 Displays—89**
 - 10.1 LCD-displays met 16 pinnen 89
 - 10.2 LCD-displays 91
 - 10.2.1 LCD-display 1602 en 2004 91
 - 10.2.2 Tekst weergeven op LCD-display 91
 - 10.2.3 Kamertemperatuur en relatieve vochtigheid op LCD-display 94
 - 10.3 Oefeningen 97
 - 10.3.1 Oefening – Backlight 97
 - 10.3.2 Oefening – Vergelijk sensoren 97
 - 10.3.3 Oefening* – Verschil 97

- 11 Actuatoren—98**
 - 11.1 Relais 98
 - 11.2 Transistor 99
 - 11.3 Relais schakelen met transistor 103
 - 11.4 FET 105
 - 11.5 PWM 106
 - 11.5.1 Frequentie en duty cycle 106
 - 11.5.2 Duty cycle van PWM regelen met potentiometer 107
 - 11.5.3 PWM op de oscilloscoop 109

11.6	Parallel proces – Blink Without Delay - Duo	110
11.7	Oefeningen	113
	11.7.1 Oefening – LED-dimmer	113
	11.7.2 Oefening – Lichtdimmer	113
	11.7.3 Oefening* – Lichtdimmer in balans	113
	11.7.4 Oefening – Blink Without Delay - Trio	113
12	Elektromotor—114	
12.1	Elektromotor schakelen met relais	114
12.2	Elektromotor schakelen en toerental regelen met FET	115
12.3	Servomotor	118
	12.3.1 Werking servomotor	119
	12.3.2 Servomotor aansturen met Arduino	121
12.4	Stappenmotor	123
	12.4.1 Werking van de stappenmotor	123
	12.4.2 Soorten stappenmotoren	125
	12.4.3 Stappenmotor aansturen met Arduino	128
	12.4.4 Stappenmotoraansturing zichtbaar gemaakt	132
12.5	Oefeningen	133
	12.5.1 Oefening – Ventilator en klep regelen met een DHT11 en servomotor	133
	12.5.2 Oefening – Airco sweeper	134
	12.5.3 Oefening – Stuurinrichting met stappenmotor	134
13	Bussystemen—135	
13.1	Seriële bus, synchroon, one-wire, één sensor	136
13.2	Seriële bus, synchroon, one-wire, een of méér sensoren	137
	13.2.1 Digitale DS18B20 temperatuursensor	137
	13.2.2 Meerdere DS18B20-temperatuursensoren op one-wire-bus	139
13.3	I2C-bussysteem	144
	13.3.1 BMP180-barometer via I2C	144
	13.3.2 - I2C en LCD-display	148
13.4	SPI (Serial Peripheral Interface)	152
13.5	UART (Universal Asynchronous Receiver & Transmitter)	153
14	Data loggen—154	
14.1	Data loggen op SD-kaart	154
14.2	Oefeningen	159
	14.2.1 Oefening – Relatie temperatuur en licht	159

15	Telemetrie en telematica—161
15.1	Telemetrie en telematica in de praktijk 162
15.2	Datatransmissie via 433 MHz 163
15.2.1	<i>Zenden op 433 MHz</i> 163
15.2.2	<i>Ontvangen op 433 MHz</i> 167
15.2.3	<i>Oefening – Professioneel weerstation via 433 MHz</i> 170
15.3	Datatransmissie via het internet 170
15.3.1	<i>Ethernet Shield W5100</i> 171
15.3.2	<i>Sensorwaarden monitoren op webpagina</i> 174
15.4	WiFi 178
15.4.1	<i>Installatie ARDUINO UNO WIFI REV2</i> 179
15.4.2	<i>WiFi-toegang en processturing</i> 180
15.5	Bluetooth 184
15.5.1	<i>Bluetooth-module HC-06</i> 184
15.5.2	<i>Licht schakelen met smartphone</i> 184
15.5.3	<i>Verkeerslicht schakelen met smartphone</i> 187
15.6	Oefeningen 190
15.6.1	<i>Oefening – E-Health</i> 190
15.6.2	<i>Oefening – Analoge regeling met Bluetooth-‘remote’</i> 190
15.6.3	<i>Oefening* – Arduino Webserver met beeld</i> 191
15.6.4	<i>Oefening* - WiFi en procesregeling via website</i> 191
16	Overige sensoren—192
16.1	Ultrasonische afstandsmeter 192
16.2	Geluidssensor 193
16.3	Hall-effect-sensoren 193
16.3.1	<i>Hall-effect</i> 193
16.3.2	<i>Hall-effect-sensor 49E en A3144</i> 195
16.3.3	<i>Toerentalmeting</i> 196
16.4	Vochtgehaltesensor 200
16.5	Regensensor 201
16.6	Windsnelheidsmeter (anemometer) 201
16.7	Shield 202
16.8	Hartslagsensor 203
16.9	GPS 206
16.9.1	<i>NEO-6M GPS sensor</i> 206
16.9.2	<i>GPS-gegevens ophalen van NEO-6M GPS</i> 208
16.9.3	<i>Positie weergeven op webpagina</i> 210
16.10	Oefeningen 212
16.10.1	<i>Oefening – Afstandsmeting en alarm met ultrasonische afstandsmeter</i> 212

- 16.10.2 *Oefening – VU-meter* 212
- 16.10.3 *Oefening* – Weerstation voltooien* 214
- 16.10.4 *Oefening* - Hall-effect-sensor-toerenteller* 214

17 Arduino-programmeertaal—216

- 17.1 *Structuur* 216
 - 17.1.1 *Structuur – control* 217
 - 17.1.2 *Structuur – divers* 221
 - 17.1.3 *Structuur – rekenkundige bewerkingen* 222
- 17.2 *Waarden – variabelen en constanten* 223
 - 17.2.1 *Variabelen* 223
 - 17.2.2 *Typen variabelen* 224
 - 17.2.3 *Constanten* 226
- 17.3 *Functies* 227
 - 17.3.1 *Digitale input- en outputfuncties* 228
 - 17.3.2 *Analoge input- en outputfuncties* 229
 - 17.3.3 *Tijdfuncties* 230
 - 17.3.4 *Wiskundige en goniometrische functies* 231
 - 17.3.5 *Random waarden* 233
 - 17.3.6 *Datacommunicatie* 233

Eindoefening—235

Appendix 1 Troubleshooting—236

- A1.1 *Hardware – algemeen* 236
- A1.2 *Hardware – voedingsspanning* 236
- A1.3 *Hardware – spanning op analoge/digitale input* 237
- A1.4 *Hardware – GND* 237
- A1.5 *Hardware – MQ-x-gassensoren* 237
- A1.6 *Hardware – Ethernet Shield W5100* 238
- A1.7 *Software – Installatiefouten* 239
- A1.8 *Software – Foutmeldingen* 239
- A1.9 *Software – Aanhalingstekens* 242
- A1.10 *Software – Onrealistische sensorwaarden* 243

Appendix 2 Datasheets—244

- A2.1 *Datasheet weerstanden* 244
- A2.2 *Datasheet NPN-Transistor BC546..560* 244
Switching and Applications 244
- A2.3a *Datasheet N-Channel FET BUZ11* 246
- A2.3b *Datasheet N-Channel FET 4N60 en 5N60* 246

- A2.4 Datasheet LDR T9-serie 247
- A2.5a Datasheet DHT11 Sensor 247
- A2.5b Datasheet RGB tri-color LED 248
- A2.6 Datasheet LM35 Sensor 249
- A2.7 Datasheet MQ3 Gassensor 249
 - A. *Standard work condition* 250
 - B. *Environment condition* 250
 - C. *Sensitivity characteristic* 250
 - D. *Structure and configuration, basic measuring unit* 250
 - E. *Electric parameter measurement circuit* 251
 - F. *Sensitivity characteristic curve of MQ-3* 251
 - F. *Sensitivity adjustment* 252
- A2.8 Datasheet Drukknopschakelaar 252
- A2.9 Datasheet DS18B20 253
- A2.10 Voorbereidende soldeerinstructies displays 253

Appendix 3 ASCII-tabel—254

Appendix 4 Onderdelenlijst—255

- Basisset Arduino vierde druk* 255
- Aanvullingsset Arduino vierde druk* 256
- Basisset opmerkingen* 256

Register—257

Inleiding

Voor je ligt een boek waarin je op een praktische manier leert hoe je de Arduino moet programmeren en kunt toepassen in de praktijk. Het is de bedoeling dat je de grenzen van deze minicomputer opzoekt en vooral, dat je die grenzen passeert. Daar hoeft je echt niet veel moeite voor te doen. Want uit ervaring weet ik dat het leren snel overgaat in doen. Je zult merken dat er zoveel inspirerende en innoverende ideeën komen opborrelen, dat het werken met de Arduino bijna verslavend wordt. Laat bijvoorbeeld een verlicht reclamebord dansen op de maat van de muziek, maak een robot die anderen doet verbazen, laat Knight Rider herleven met lopende LEDs, bedenk het maar. Met de Arduino is echt alles mogelijk, dus overstijg vooral jezelf! Succes!

Jacco de Jong

Toelichting bij derde druk

De derde druk is aangepast aan de laatste stand van de techniek. Nieuw in de derde druk zijn GPS plaats- en snelheidsbepaling, hartslagmeting en -monitoring, multicolor-LED, contactloze toerentalmeting via een magnetisch veld (Hall-effect-sensor), bussystemen en WiFi op basis van de optionele Arduino UNO WIFI Rev2. Bussystemen worden steeds vaker toegepast en vereenvoudigen het bedradingswerk. Traditioneel wordt elke sensor met twee of drie draden aangesloten op de Arduino, dus met vijf sensoren zijn dat al maximaal 15 draden. Door toepassing van een bus zijn slechts twee draden nodig, waar alle sensoren op aangesloten worden. Omdat elke sensor een uniek adres heeft, weet de Arduino het juiste commando naar de juiste sensor of actuator te sturen.

Leerdoelen

- installeren van de Arduino-software
- kennis van programmeertaal
- bouwen en aanpassen van Arduino-sketch
- hergebruik van programmacode
- kennis van analoge en digitale elektronica
- kennis van actieve en passieve elektronische componenten
- kennis van mechatronische componenten
- kennis van seriële communicatie en LCD-displays
- toepassen van hardware en software in projecten
- data loggen
- datatransmissie en telemetrie

Extra materiaal

Voor de docenten is extra materiaal verkrijgbaar, onder andere de gezipte sketches, via <https://boomberoepsonderwijs.nl/lesmethode/brinkmanict-info/> optie *Docent Service*.

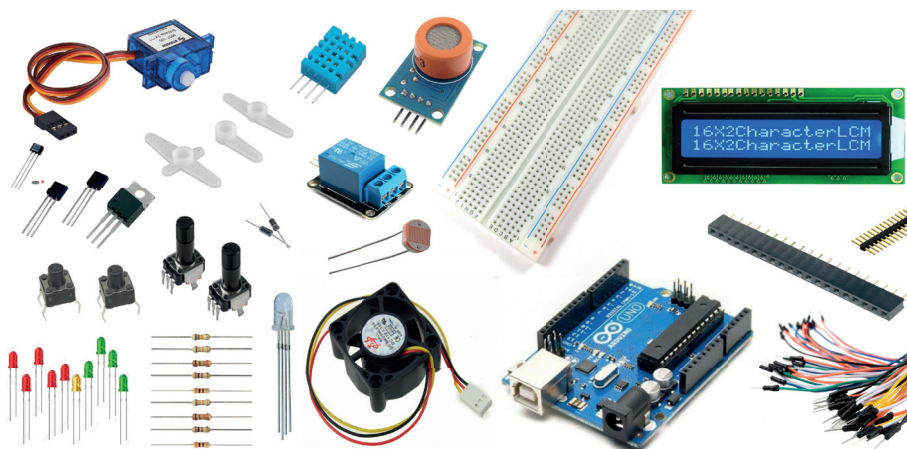
Hier worden ook eventuele errata gepubliceerd.

1 Aftrap

De Arduino is ooit ontworpen voor ontwerpers en kunstenaars in de Italiaanse mode-industrie: de kleine computer moest LED-verlichting in kleding aansturen. Inmiddels is de Arduino een breed toegepaste microcontroller met een behoorlijke 'crashbestendigheid'. Hij is gemakkelijk te doorgronden en ideaal als leerobject. Sterker nog, voor velen is de Arduino verslavend: hij nodigt uit tot meer en tot het verleggen van grenzen. De toepassingsmogelijkheden van de Arduino zijn dan ook bijna grenzeloos! Bovendien is het programmeren van een Arduino niet moeilijk. Je hebt er weinig of geen programmeerervaring voor nodig.

1.1 Aanschaf

Om met dit boek te kunnen werken heb je een Arduino nodig. Je kunt er een aanschaffen in een winkel of online. Prijzen variëren van 10 tot 25 euro. Het is aan te raden om een complete basisset aan te schaffen met daarin alle onderdelen uit de Onderdelenlijst van Appendix 4.



Afbeelding 1.1 – Voorbeeld van een basisset met sensoren en actuatoren.

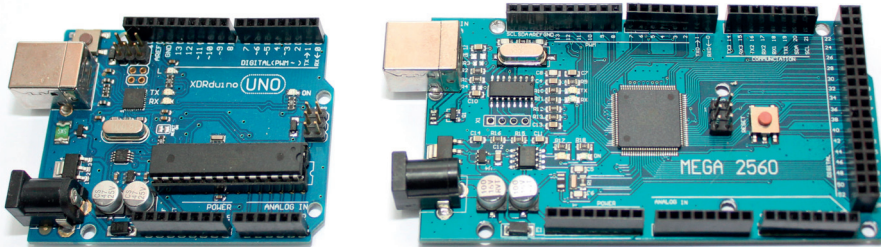
In de basisset vind je diverse sensoren en actuatoren. In vrijwel elk Arduino-project kom je die tegen. Vrij vertaald zijn het 'voelers' en 'doeners'. Een temperatuursensor voelt bijvoorbeeld hoe heet of hoe koud het is. Afhankelijk van de gemeten temperatuur stuurt de Arduino een actuator aan. Die actuator schakelt dan bijvoorbeeld de verwarming aan of uit.

Een sensor is een elektrisch ding dat een natuurkundige grootheid *meet*, zoals temperatuur, relatieve vochtigheid, druk, lichtintensiteit, gasconcentratie, straling enzovoort. Een actuator is een elektrisch ding dat iets *doet*. Zoals een LED (geeft licht), een motor (draait rond), een elektromagnetische klep (kan open of dicht), een relais (zet verwarming aan of uit) enzovoort. Afhankelijk van je project bepaal je welke sensor(s) en actuator(en) je nodig hebt.

1.2 Arduino UNO R3 of Arduino MEGA 2560?

De Arduino wordt in meerdere uitvoeringen geleverd. De bekendste zijn Arduino UNO R3 en de duurdere Arduino MEGA 2560. Beide worden geleverd als print of 'board' met daarop een paar onderdelen. In een voltooid project wordt de compacte print gewoonlijk ingebouwd in een kastje of in een apparaat, maar in dit boek werk je met het 'kale' board.

Alle hier beschreven voorbeelden zijn gebaseerd op de Arduino UNO R3, maar werken ook op de Arduino MEGA 2560. De Arduino MEGA 2560 kan hetzelfde als de UNO R3, maar heeft meer in- en uitgangen voor sensoren en actuatoren. In latere projecten kunnen deze extra in- en uitgangen goed van pas komen, voor dit boek maakt het niet uit welke van de twee Arduino's je kiest. Waar nodig worden in dit boek de verschillen tussen de beide boards toegelicht; dat is onder andere het geval bij het instellen van de software.



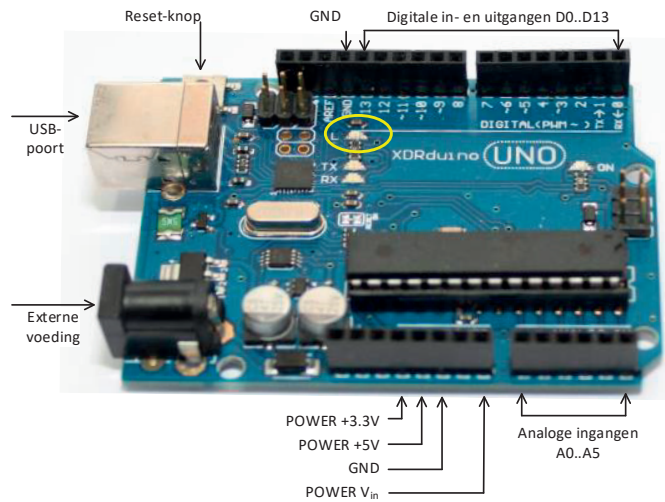
Afbeelding 1.2 -Twee Arduino-boards: links de Arduino UNO R3 en rechts de Arduino Mega 2560.

Arduino UNO WIFI Rev2

Wil je aan de slag met het hoofdstuk over WiFi dan heb je een Arduino UNO WIFI Rev2 nodig. Deze versie van de Arduino is nog vrij nieuw en kostbaar en zit niet in de Basisset. Het hoofdstuk over WiFi is dan ook optioneel. Wil je met WiFi aan de slag schaf dan ook de Arduino UNO WIFI Rev2 aan.



1.3 Poorten en voeding



Afbeelding 1.3 – Aansluitingen en poorten van de Arduino UNO R3. Led L is geel omcirkeld.

Heb je je keuze gemaakt, maak jezelf dan eerst vertrouwd met het board. Afbeelding 1.3 toont het board van de Arduino UNO R3. Rechts naast de USB-poort zie je de (vierkante) microprocessor. Verder zie je vooral poorten, zoals ingangen (voor het aansluiten van sensoren) en uitgangen (voor het aansluiten van actuatoren). Om te kunnen functioneren heeft de Arduino voeding nodig. Die voeding krijgt hij van je laptop of pc via de USB-poort; de poort 'Externe voeding' hoeft je dan niet te gebruiken (zie kader).

Aanduiding poort	Omschrijving
USB	USB-poort voor computeraansluiting
Externe voeding	Batterij of netvoeding (6 tot 20 volt)
POWER +3.3V	3,3 volt uitgang voor externe sensoren
POWER +5V	5 volt uitgang voor externe sensoren
POWER V _{in}	9 of 12 volt uitgang, afhankelijk van 'Externe voeding'
DIGITAL Do..D13	Digitale in- en uitgangen (14 pinnen)
ANALOG IN Ao..A5	Analoge ingangen (6 pinnen)
GND	0 volt of massa (ground)

Tabel 1.1 – Belangrijkste poorten van de Arduino UNO R3

Externe voeding

Om los van de laptop of pc te kunnen functioneren kan de Arduino via de poort ‘Externe voeding’ worden gevoed met een 9 volt blokbatterij of een netadapter van 5 tot 12 volt. Volt wordt vaak afgekort tot de letter V of VDC. Ook een kleine 12 volt accu is geschikt; gebruik in dat geval altijd een zekering van circa 1 ampère. Als je een externe voeding gebruikt, wordt de voeding via USB automatisch uitgeschakeld. Je kunt de USB-kabel dus gewoon laten zitten.

1.4 Systeemeisen

De Arduino moet geprogrammeerd worden. Dat doe je op je pc of laptop met de Arduino-software. De Arduino-software draait onder Windows, Mac OS X 10.7 Lion of later, Linux 32-bits, Linux 64-bits en Linux ARM. In dit boek wordt uitgegaan van Windows 10, maar Arduino-software draait op elke x86- en x64-versie van Windows vanaf Windows XP.

1.5 Software

De Arduino-software is open source en gratis te downloaden van:

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

In een paar stappen zet je de Arduino-software op je laptop of pc. Als voorbeeld is gekozen voor een systeem met Windows 10, omdat dit het meest gebruikte besturingssysteem is.

NB Computers op scholen, in openbare ruimtes en bij bedrijven zijn vaak zo geconfigureerd, dat alleen een beheerder nieuwe software mag installeren. Mogelijk dat het je wel lukt om software op zo’n computer te installeren, maar het kan zijn dat de computer – na het opnieuw opstarten – weer terugvalt in zijn basisconfiguratie. Dat wil zeggen dat de geïnstalleerde software en bijbehorende projecten verdwenen zijn. Informeer hiernaar voordat je de software installeert.

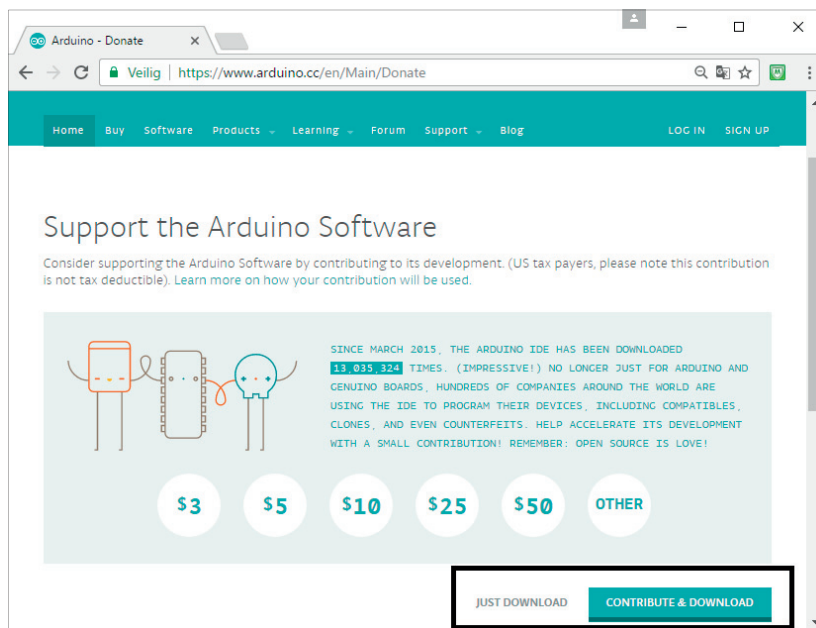
ACTIE

- Start je browser en ga naar:
<https://www.arduino.cc/en/Main/Software>
Selecteer het te installeren besturingssysteem. In dit geval *Windows Installer* (omkaderd in afbeelding 1.4).



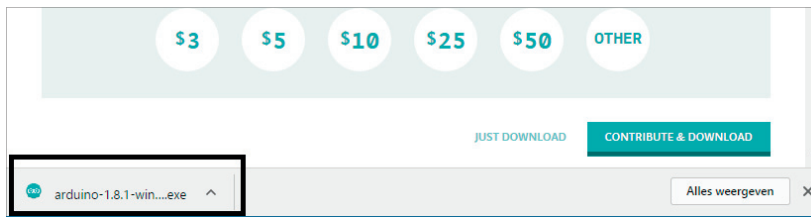
Afbeelding 1.4 – Keuze besturingssysteem.

- De software is in principe gratis, maar een vrijwillige donatie wordt op prijs gesteld (zie afbeelding 1.5). Met een donatie draag je bij aan het verder ontwikkelen van de software. Wil je niet doneren, klik dan rechts onderin op *Just download*.



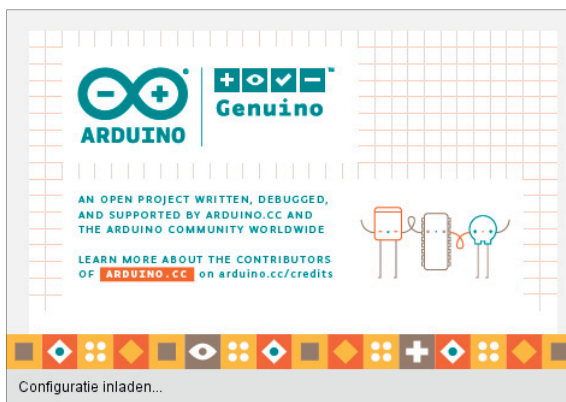
Afbeelding 1.5 – Vrijwillige bijdrage.

- Linksonder kun je het downloadproces volgen (zie omkadering in afbeelding 1.6). Zodra de download is voltooid klik je op de bestandsnaam om het installatieproces te starten.

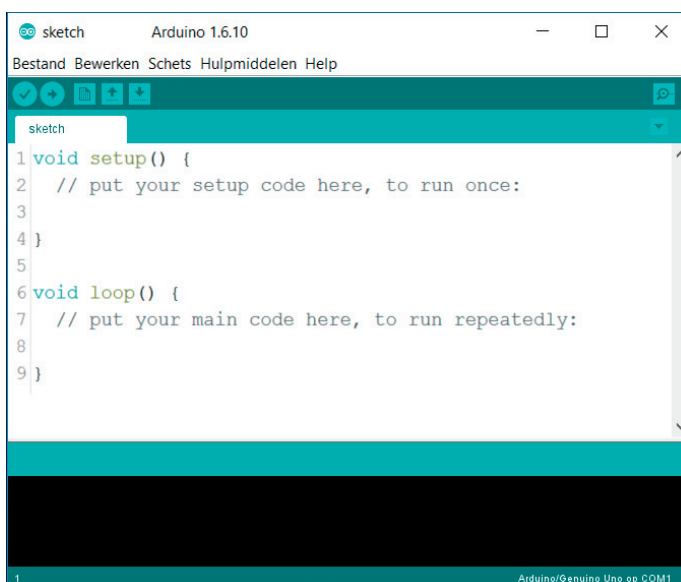


Afbeelding 1.6 – Het te installeren bestand is gedownload.

- Na de installatie zal het programma automatisch starten en zichzelf configureren (zie afbeeldingen 1.7 en 1.8). Mogelijk wordt ook gevraagd of hulpprogramma's (zoals USB-stuurprogramma's) geïnstalleerd mogen worden. Beantwoord die vragen met *Ja*. Na het configureren sluit het programma automatisch.



Afbeelding 1.7 – Starten van Arduino-software.



Afbeelding 1.8 – Openingscherm van Arduino-software.

- Start de Arduino-software nu zelf. Dat kan op verschillende manieren, bijvoorbeeld (zie afbeelding 1.9):
 - Klik op de Windows *Start*-knop en klik in de lijst op *Arduino*.
 - Klik op de icoon of tegel op het bureaublad.



Afbeelding 1.9 – Opstarten via Windows Start.

Telkens na het starten zal de Arduino-software op zoek gaan naar software-updates. Daartoe maakt de software verbinding met het internet. Mogelijk krijg je dan een melding van de Windows Firewall. Klik in dat geval op *Toegang toestaan*.

- Je moet nu nog twee dingen doen: hardware aansluiten en software instellen, zie volgende paragrafen.

1.6 Arduino met pc verbinden

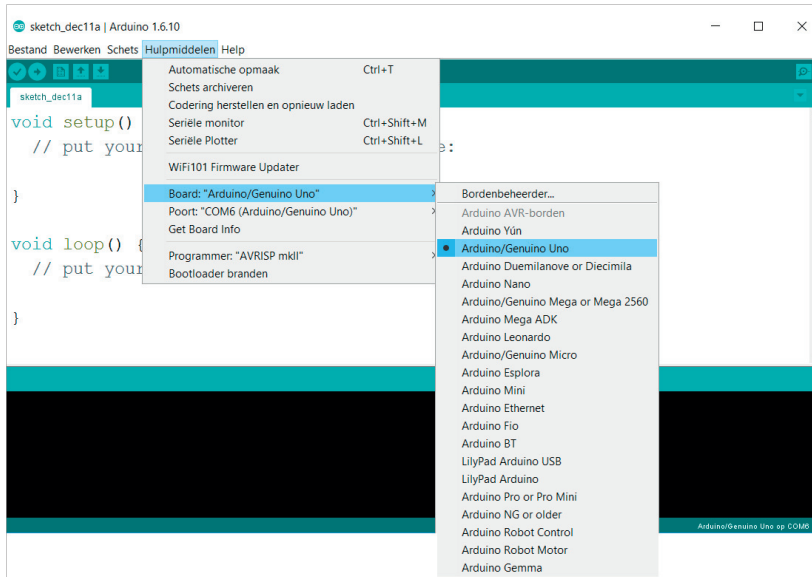


Afbeelding 1.10 – USB-verbinding tussen Arduino UNO en laptop.

Om de Arduino vanaf je pc of laptop te programmeren, moeten ze met elkaar kunnen communiceren. Daartoe verbind je de Arduino via een USB-kabel met je pc of laptop. Omdat de Arduino via die kabel ook zijn voeding krijgt, licht de ON-LED op het Arduino-board op. Enkele seconden later is ook de verbinding met de computer tot stand gekomen. Mogelijk wordt tijdens de eerste keer dat je de software gebruikt een USB-stuurprogramma gedownload en geïnstalleerd, dit kan enkele minuten duren.

NB Twee andere LEDs die mogelijk oplichten zijn TX en RX en hebben te maken met datacommunicatie via de USB-poort of een andere communicatiepoort.

1.7 Software instellen



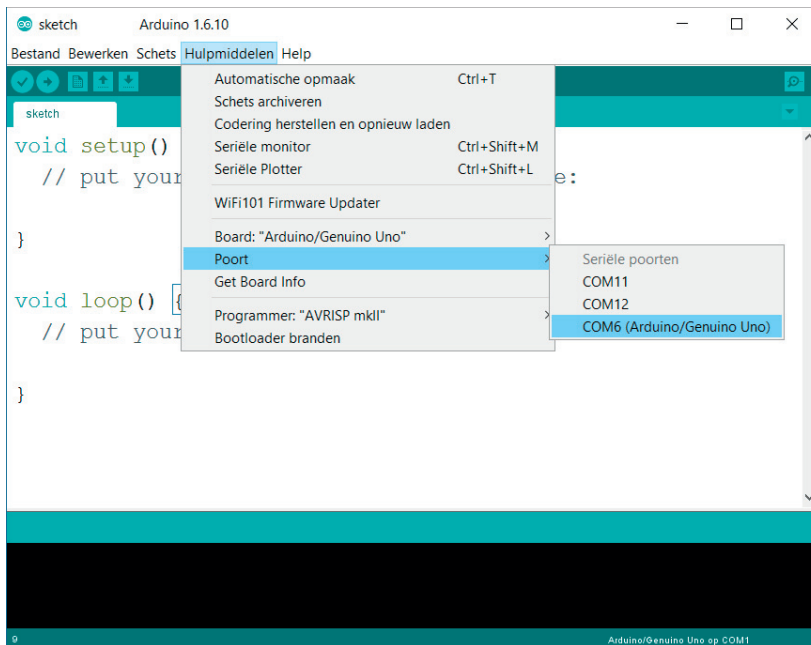
Afbeelding 1.11 – Instellen van de juiste Arduino.

Na het installeren van de Arduino-software en het aansluiten van de hardware moeten drie instellingen in de software worden gemaakt:

- instellen type Arduino;
- instellen COM-poort om met de Arduino te communiceren;
- inschakelen regelnummering.

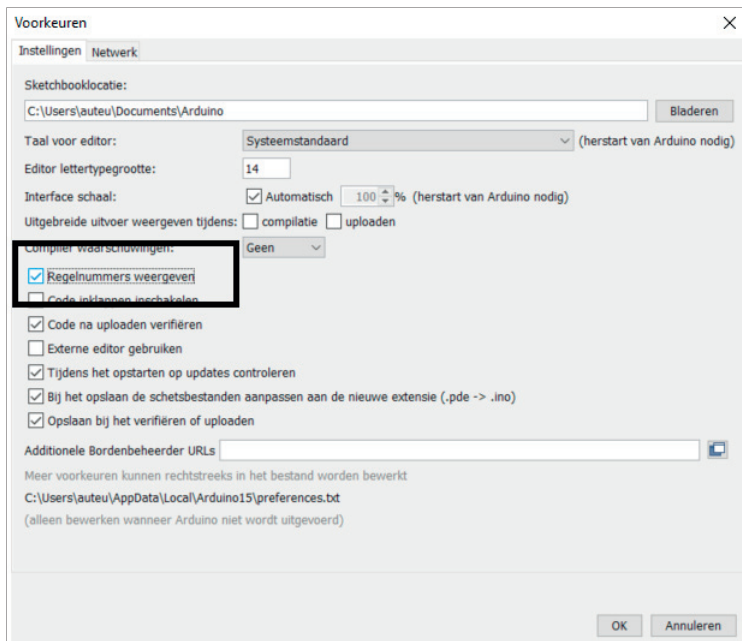
Start de Arduino-software, open het menu *Hulpmiddelen* en kies *Board*. Je ziet nu een lijst van verschillende Arduino's (zie afbeelding 1.11). Selecteer *Arduino/Genuino UNO*. Heb je een Arduino MEGA 2560, selecteer dan *Arduino/Genuino Mega or Mega 2560*.

Klik vervolgens in het menu *Hulpmiddelen* op *Poort* (zie afbeelding 1.12). Doorgaans is de juiste poort al ingesteld. In het afgebeelde voorbeeld is dat COM6. Dit betekent dat de USB-poort zich gedraagt als 'COM'unicatie-poort nummer 6. Deze instelling is vooral bedoeld voor wie twee of meer Arduino's op de computer heeft aangesloten: elk heeft dan zijn eigen COM-poort en in dit menu kies je welke Arduino je met de software wilt aansturen.



Afbeelding 1.12 – Instelling van de juiste COM-poort.

De derde instelling betreft het inschakelen van de regelnummers in de sketches. Regelnummers zijn handig, bijvoorbeeld bij het lokaliseren van fouten. Je zult dat later zien. Open het menu *Bestand* en klik op *Voorkeuren* (zie afbeelding 1.13). Zet een vinkje bij de optie *Regelnummers weergeven* en sluit het venster.



Afbeelding 1.13 – Voorkeuren instellen.

2 Eerste project

In dit hoofdstuk maak je kennis met de Arduino-programmeertaal. Met de programmeertaal maak je een programma of ‘schets’, waarin je de Arduino vertelt wat hij moet doen. Je maakt de schets op je pc en ‘stuurt’ hem daarna naar de Arduino. In plaats van de Nederlandse termen ‘schets’ en ‘sturen’ zul je vaker de Engelse termen tegenkomen: ‘sketch’ en ‘upload’. In dit hoofdstuk maak en upload je een sketch die een LED laat knipperen.

2.1 Knipperende LED

Op het Arduino-board zitten vier LEDs. Wat een LED precies is komt later ter sprake. Een van de vier LEDs brandt al: deze LED is gemarkeerd *ON*. Een andere LED is aangesloten op de digitale uitgang D13 en is gemarkeerd *L*. In afbeelding 1.3 is deze LED geel omcirkeld. Met een simpele sketch kun je deze LED laten knipperen of (in het Engels) ‘blink’. De sketch die je hiervoor nodig hebt heet **Blink** en is al als voorbeeldsketch op je pc aanwezig.

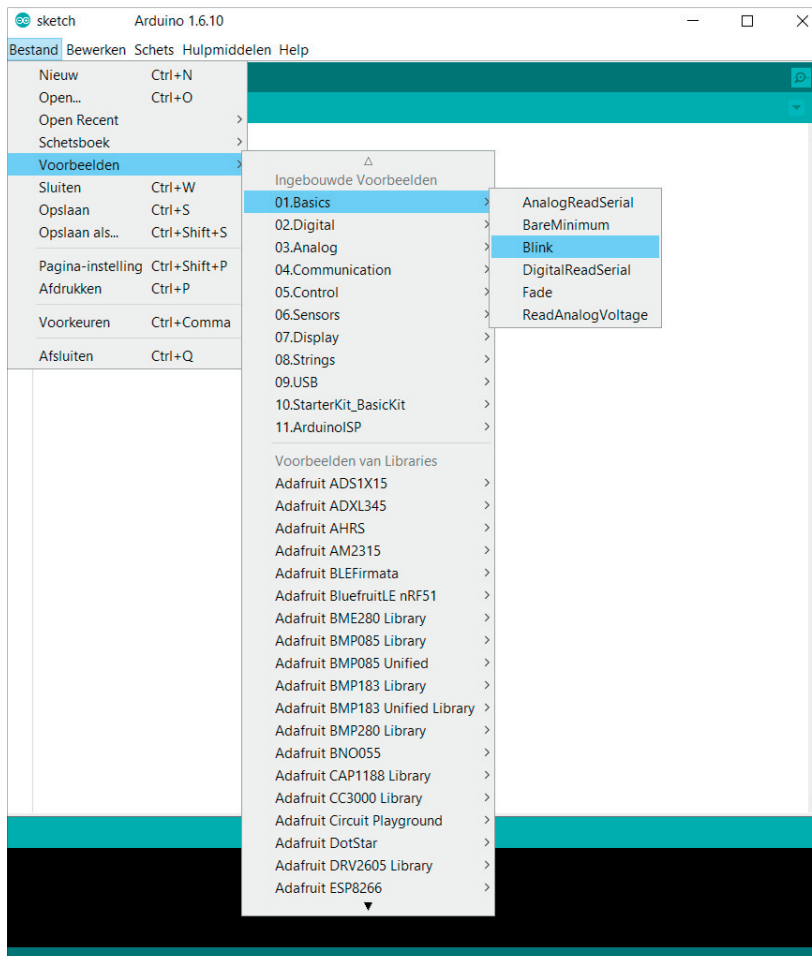
ACTIE

- Zorg dat er geen losse onderdelen rond de Arduino liggen. Als die per ongeluk onder het Arduino-board terecht komen kunnen ze kortsluiting veroorzaken en de elektronica beschadigen.
- Verbind de Arduino via de USB-kabel met je pc.
- Start de Arduino-software.
- Kies de sketch: open het menu *Bestand > Voorbeelden > 01.Basics > Blink* (zie afbeelding 2.1).
- Vervolgens opent een nieuw venster met daarin de sketch (zie afbeelding 2.2). De afgebeelde sketch bevat Nederlandstalige uitleg, op het scherm is de uitleg in het Engels.
- In de groene menubalk van het venster zie je een paar knoppen. Beweeg er met je muis overheen om de betekenis van de knoppen te lezen. Klik op de knop *Upload*.
- Een groene voortgangsbalk geeft het compileren en uploaden van de sketch naar de Arduino weer. Compileren is het vertalen van de sketch naar de machinetaal van de microprocessor op het Arduino-board.
- Tijdens het uploaden knipperen op het Arduino-board de communicatie-LEDs TX en RX. Wacht tot de LEDs TX en RX niet meer knipperen.
- De sketch draait nu: de LED op D13 knippert 1 seconde aan, 1 seconde uit en zo voort.

Tijdens het uploaden wordt de sketch in het geheugen van de Arduino geplaatst. Dit geheugen is flash-geheugen dus de sketch blijft bewaard, ook als de voeding wegvalt. Probeer dit uit:

ACTIE

- Trek de USB-kabel uit de Arduino, de LED dooft.
- Wacht een paar seconden en steek de USB-stekker er weer in: de sketch start opnieuw en de LED knippert weer.



Afbeelding 2.1 – Open de sketch **Blink**.


```

1 // Blink (knipperen) Zet de LED gedurende 1 seconde (1000 ms) aan en daarna 1 seconde
2 // Op nagenoeg iedere Arduino zit wel een LED die je kunt laten knipperen.
3 // Vertaald 09-12-2016 door J.J. de Jong. */
4
5 // void setup wordt eenmalig geactiveerd zodra:
6 // - de Arduino wordt ingeschakeld
7 // - de Reset-knop wordt ingedrukt
8
9 void setup() {
10 // initialiseren van digital 13 als uitgang (output).
11 pinMode(13, OUTPUT);
12 }
13 // de loop wordt continu herhaald.
14 void loop() {
15 digitalWrite(13, HIGH); // Zet de LED op pin 13 aan (HIGH betekent 5 Volt op pin 13).
16 delay(1000); // Wacht een seconde (1000 ms)
17 digitalWrite(13, LOW); // Zet de LED op pin 13 uit (LOW betekent 0 Volt op pin 13).
18 delay(1000); // Wacht een seconde (1000 ms)
19 }

```

Compilieren voltooid.

De schets gebruikt 928 bytes (2%) programma-opslagruimte. Maximum is 32256 bytes.
 Globale variabelen gebruiken 9 bytes (0%) van het dynamisch geheugen. Resteren 2039 bytes

3 Arduino/Genuino Uno op COM5

Afbeelding 2.2 – Sketch (programma) voor knipperende LED.

2.2 Sketch

In dit hoofdstuk wordt de structuur van een sketch uitgelegd. Elke sketch bestaat uit minimaal twee functies: *void setup* en *void loop*. In afbeelding 2.2 zijn die blauw omkaderd. Tussen de accolades staan statements. Een statement is een regel met code en aan het eind een puntkomma. Een statement bevat onder meer variabelen en andere functies. In de *void setup* zie je bijvoorbeeld een statement met de functie *pinMode()*:

```
pinMode(13, OUTPUT);
```

In *void loop* zie je onder andere een statement met de functie *delay()*:

```
delay(1000);
```

De Arduino-programmeertaal is hoofdlettergevoelig. Let dus goed op de juiste notatie van de hoofdletter in bijvoorbeeld *pinMode*. Spaties daarentegen hebben geen betekenis voor de programmeertaal. Je kunt ze gebruiken om je sketch visueel en overzichtelijk te ordenen. Dat kan op verschillende manieren, zoals de sketches in dit boek laten zien.

Als geheugensteun voor jezelf en om andere programmeurs uit te leggen hoe de sketch werkt, maak je gebruik van commentaarregels. Daar zijn twee manieren voor:

- Een commentaar van meerdere regels zet je tussen de tekens `/*` en `*/`. Zie bijvoorbeeld de eerste drie rood omkaderde regels in afbeelding 2.2. De tekens zijn groen omcirkeld. Tekstregels zijn alleen voor de programmeur zichtbaar en worden niet geüpload naar de Arduino. Ze zullen dus geen geheugenruimte van de Arduino in beslag nemen.
- Gaat het om één enkele commentaarregel (zoals regel 5) begin dat commentaar dan met de tekens `//` (groen omcirkeld). Er zijn geen eindtekens, want het einde van de regel is automatisch het eindteken. Ook deze commentaarregels worden niet naar de Arduino geüpload.

Alles wat tussen de accoladetekens van de functie `void setup` staat wordt één keer uitgevoerd: telkens als het programma opstart of nadat op de Reset-knop op het Arduino-board is gedrukt. In de `void setup` bepaal je onder andere welke pinnen je als in- en uitgangen gaat gebruiken. In het voorbeeld wordt pin D13 met de functie `pinMode` als `OUTPUT` (uitgang) ingesteld. Je zult zien dat de Arduino-programmeertaal op 'vreemde' plaatsen hoofdletters gebruikt, zoals de `M` in `pinMode`. De bedoeling daarvan is om duidelijk onderscheid te maken tussen twee woorden: pin en Mode. Om dezelfde reden zie je in afbeelding 2.2 dat tekst, commando's en waarden in verschillende kleuren worden weergegeven.

```
void setup() {
// Initialiseer digitale poort D13 als uitgang (OUTPUT):
pinMode(13, OUTPUT);
}
```

Alles wat tussen de accoladetekens van de functie `void loop` staat (tweede blauwe omkadering in afbeelding 2.2) wordt continu herhaald. Loop betekent hier 'herhaallus'. In elke herhaallus worden door de Arduino-processor statements uitgevoerd en waarden gelezen. In het voorbeeld is dat het sturen van een HIGH of LOW naar uitgang 13 en daarna 1 seconde wachten. HIGH betekent hier 5 volt op pin 13, LOW betekent 0 volt op pin D13. De precieze betekenis van de functies en waarden lees je in het volgende hoofdstuk.

```
// de loop wordt continu herhaald.
void loop() {
digitalWrite(13, HIGH); // Zet de LED op pin 13 AAN
delay(1000);           // Wacht een seconde (1000 ms)
digitalWrite(13, LOW); // Zet de LED op pin 13 UIT
delay(1000);           // Wacht een seconde (1000 ms)
}
```

Samenvattend kun je de sketch **Blink** als volgt lezen: de sketch begint met het definiëren van uitgang 13 als digitale output (regel 9 in afbeelding 2.2). In regel 13 begint de herhaallus. Met het commando 'zet uitgang 13 HOOG' wordt de LED op pin D13 AAN gezet (regel 15). Daarna begint de functie `delay(1000)`, ofwel 'wacht

1000 milliseconden' (regel 16). Tijdwaarden worden in de functie *delay* uitgedrukt in milliseconden (1000 milliseconden is 1 seconde). Na de pauze stuurt de microprocessor het commando 'zet uitgang 13 LAAG' (regel 17). De LED is nu uit. Daarna wordt het volgende statement *delay(1000)* gelezen (regel 18): 'wacht 1000 milliseconden.' Na de pauze wordt de loop herhaald. De microprocessor stuurt weer het commando 'zet uitgang 13 HOOG' (regel 15), enzovoort.

2.3 Functies en waarden

Functies en waarden in de sketch bepalen wat de Arduino moet doen. De Arduino-programmeertaal kent tientallen functies en waarden, waarvan je de meeste in dit boek zult leren kennen. In de voorbeeldsketch zijn de volgende functies en waarden (constanten en variabelen) gedefinieerd en gebruikt:

► **pinMode (poort, OUTPUT / INPUT)**

Elk van de digitale poorten Do..D13 kan fungeren als ingang of als uitgang. Het is zaak dat je voor elke digitale poort die je in je sketch gebruikt, aangeeft hoe die poort moet werken. Dat doe je met de functie *pinMode*. In de functie noteer je het nummer van de poort en de modus waarin die poort moet werken: als INPUT of als OUTPUT. INPUT en OUTPUT zijn constanten, ze zijn dus niet variabel. In de modus 'ingang' kun je er sensoren op aansluiten, zoals een schakelaar of temperatuursensor. In de modus 'uitgang' kun je er actuatoren op aansluiten, zoals een LED, een motor, een elektromagnetische klep enzovoort.

```
pinMode(13, OUTPUT);
```

► **digitalWrite (poort, HIGH / LOW)**

Deze functie stuurt een digitale poort aan. Tussen haakjes staan het nummer van de poort en de waarde die de bijbehorende pin krijgt: HIGH of LOW. HIGH en LOW zijn constanten, ze zijn dus niet variabel. LOW betekent digitaal 0 (nul), HIGH betekent digitaal 1. Zo wordt met *digitalWrite(13, HIGH)* pin D13 HOOG gezet, waardoor de daarop aangesloten LED oplicht.

```
digitalWrite(13, HIGH);
digitalWrite(13, LOW);
```

► **delay(x)**

De functie *delay(x)* last een pauze *x* in. Tijdwaarden worden in de functie uitgedrukt in milliseconden en vermeld tussen haakjes: 1000 milliseconden is 1 seconde. Zodra de ingestelde pauze om is, gaat de Arduino verder naar het volgende statement.

```
delay(1000);
```

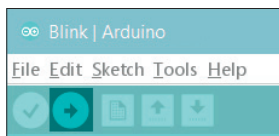
► parameters

De meeste Arduino-functies worden gevolgd door twee haakjes. Tussen die haakjes staan zogenaamde ‘parameters’. Parameters bepalen mede de uitkomst en werking van de functie. Zo zorgt de parameter ‘1000’ in de *delay*-functie voor een pauze van 1000 milliseconden. Sommige functies hebben één parameter, andere hebben er twee of meer. Afhankelijk van de functie kan een parameter een variabele zijn, een constante of tekst.

2.4 Verifieer versus Upload

Naast de *Upload*-knop zie je helemaal links in de menubalk ook een *Verifieer*-knop (knop met vinkje). De betekenis van beide knoppen is:

- *Verifieer*-knop – Hiermee kun je de sketch controleren, vóórdat je hem naar de Arduino uploadt; de verificatie geeft onder andere de grootte van de sketch aan, zodat je weet of hij wel of niet in het flash-geheugen van de Arduino zal passen. Eventuele fouten in de sketch worden onderin beeld gemeld. Corrigeer ze en herhaal de verificatie totdat de sketch foutloos is.
- *Upload*-knop – Hiermee stuur je de sketch naar de Arduino. Een eventueel reeds aanwezige sketch wordt automatisch uit het geheugen gewist. Na het uploaden start de sketch automatisch.



Afbeelding 2.3 – *Verifieer*-knop en *Upload*-knop.

Wat de harde schijf is voor je pc, is het ingebouwde flash-geheugen voor de Arduino. Het flash-geheugen van een Arduino UNO R3 is met 32 kilobytes tamelijk beperkt, temeer omdat hiervan slechts 28 kB beschikbaar is voor de sketches. De oefeningen in dit boek passen daar wel in, maar in de praktijk komen grote en complexe sketches voor, die te groot zijn voor een UNO R3. Met de *Verifieer*-knop kom je daar snel genoeg achter. De Arduino MEGA 2560 biedt dan de oplossing, want die heeft met 256 kB aanzienlijk meer ruimte.

Wat het werkgeheugen is voor je pc, is het SRAM-geheugen voor de Arduino. Dit geheugen wordt gewist zodra je de Arduino uitschakelt. Het SRAM-geheugen is 2 kB voor een UNO R3 en 8 kB voor een MEGA 2560.



Afbeelding 2.4 – Knipperen in de praktijk.

2.5 Oefeningen

In de volgende oefeningen breng je wijzigingen aan in de sketch en laat je de Arduino werken zonder USB-verbinding.

2.5.1 Oefening – Knippersnelheid aanpassen

Door de sketch aan te passen en opnieuw te uploaden kun je de Arduino andere dingen laten doen. Zo kun je bijvoorbeeld de LED sneller laten knipperen.

- Verander de knippersnelheid in de sketch van 1000 ms in 200 ms.
- Verifieer de aangepaste sketch met de *Verifieer*-knop.
- Upload de aangepaste sketch naar de Arduino en bekijk het resultaat.

2.5.2 Oefening – Sketch opslaan en openen

Er zijn tal van redenen om een sketch op te slaan: back-up, verspreiden via het internet, kopiëren naar andere Arduino's enzovoort. In deze oefening laat je afwisselend de sketch **Blink** en **Blink2** werken:

- Bewaar de gewijzigde sketch onder de naam **Blink2** via *Bestand* > *Opslaan als*. De naamextensie van een sketch-bestand is *.ino*.
- Sluit de Arduino-software via *Bestand* > *Afsluiten*.
- Open de oorspronkelijke sketch **Blink** via *Bestand* > *Open Recent* en upload de sketch.
- Open de gewijzigde sketch **Blink2** via *Bestand* > *Open Recent* en upload de sketch.

2.5.3 Oefening – Zonder USB-kabel

Zodra een Arduino via *Upload* geprogrammeerd is en de sketch in zijn flash-geheugen staat, is de computerverbinding in principe niet meer nodig. Met een externe voeding kun je de Arduino zelfstandig verder laten werken, zonder hulp van de computer.

- Trek de USB-stekker uit de Arduino. Alle LEDs op het board doven nu, want de voedingsspanning is weggevallen.

- Sluit een 9 volt blokbatterij aan op de poort 'Externe voeding' (zie afbeelding 1.3).
- De sketch wordt opnieuw opgestart en de LED knippert weer.

Oefeningen en inhoud van dit boek worden regelmatig geüpdatet met extra aanwijzingen en tips. Check daarom regelmatig de website van Boom beroepsonderwijs (<https://boomberoepsonderwijs.nl/lesmethode/brinkmanict-info/> optie *Docent Service*) en download de laatste update.
