

Internet of Things
New Media Design
Grafische en Digitale Media
Artevelde Hogeschool
2018-2019



ARDUINO EN TINKER KIT

Paulien De Fraine & Thomas Van De Velde

Inhoudstabel

Inhoudstabel.....	1
Arduino.....	2
De Arduino Uno.....	2
De Arduino software	4
Specificaties van de Arduino Uno	6
Tinkerkit.....	7
Examples.....	10
Example One Lampje laten branden	10
Example Two Lampje dimmen met de potentiometer	11
Example Three Het verkeerslicht.....	12
Example Four The Tilt, Touch en Temperature sensor.....	12

Arduino

Arduino is een open-source hardware en software bedrijf, project en gebruikersgemeenschap dat een populaire serie microcontroller-boards ontwerpt en produceert. Het produceert hiervoor ook speciale kits om digitale toestellen te bouwen en interactieve objecten met elkaar te laten interageren die zowel fysisch als digitaal zijn.

De producten vallen onder een general public license en maakt dat de productie van de Arduino boards en de software verdeling vrij is voor iedereen. De Arduino is op zich dus een minicomputer.

Het bord zelf bestaat uit een deel van digitale en analoge input en output poorten (I/O). Dit zorgt ervoor dat het bord met verschillende toestellen of toebehoren kan verbonden worden, zoals andere borden, HAT's of circuits. Om het bord aan de praat te krijgen, verbinden we het met een USB aan de computer. Zo kunnen we de software op de computer, die geschreven is in een soort combinatie van C en C++, laten reageren met het bord. Omdat we deze speciale taal niet kunnen schrijven in een normale code-editor, heeft Arduino gezorgd voor een eigen integrated development environment (IDE) die gebaseerd is op de codeertaal.

De Arduino Uno

In dit project werken met de Arduino Uno. Arduino vindt zijn begin in Italië, maar door de open-source software en de vrij gedistribueerde designs heeft het Arduino-compatible R3 UNO bord zijn oorsprong gevonden in China. Dit wel zonder het logo en zonder de “Made in Italy” tekst. Verder was dit bord identiek.



De hardware-referentieontwerpen worden gedistribueerd onder een Creative Commons Attribution Share-Alike 2.5-licentie en zijn beschikbaar op de Arduino-website. Layout- en productiebestanden voor sommige versies van de hardware zijn ook beschikbaar.

Al is dit alles vrij makkelijk verkrijgbaar, heeft Arduino zelf gevraagd om bij afgeleide producten de naam Arduino niet te vertonen op de materialen. De manier hoe ze dit omzeilen is door achter de nieuwe naam voor deze producten -duino te gebruiken of vermelden.

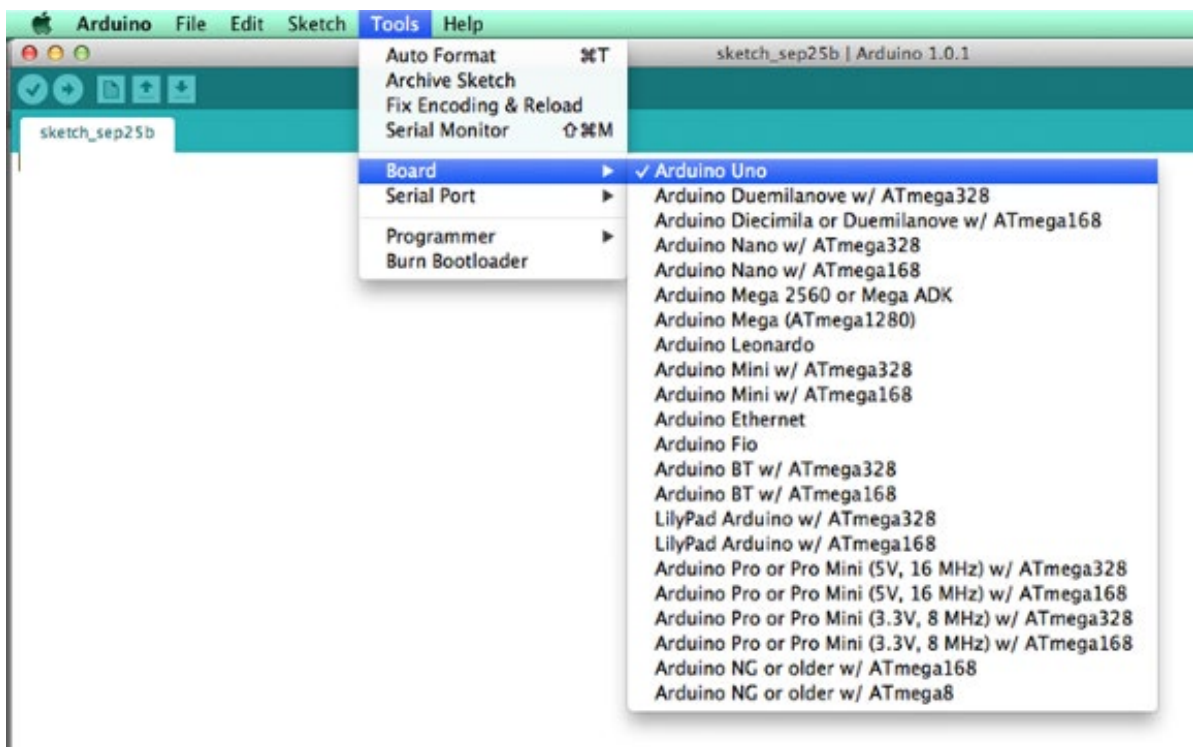
De meeste Arduino borden bestaan uit ene Atmel 8-bit AVR microcontroller. De AVR was een van de eerste microcontroller-families die gebruik maakte van on-chip flashgeheugen voor programmaopslag. De Arduino maken gebruik van een enkele of dubbele rij van vrouwelijke pinnen waarmee we verschillende verbindingen kunnen maken en andere circuits kunnen aansluiten. Deze verbindingen worden vaak gemaakt met HAT's of shields.

Verschillende Arduino-compatibele en Arduino-afgeleide kunnen onderling met elkaar uitgewisseld worden. Verschillende versterken het basis Arduino-bord zelfs door output stuurprogramma's toe te voegen.

De Arduino software

De Arduino geïntegreerde ontwikkelomgeving (IDE) is een platformonafhankelijke toepassing die is geschreven in de programmeertaal Java. Het bevat een code-editor met functies zoals tekst knippen en plakken, zoeken en vervangen van tekst, automatische inspruing, brace matching en syntax highlighting, en biedt eenvoudige one-click mechanismen voor het compileren en uploaden van programma's naar een Arduino-bord. Het bevat ook een berichtengebied, een tekstconsole, een werkbalk met knoppen voor algemene functies en een hiërarchie van bedieningsmenu's.

De Arduino IDE ondersteunt de talen C en C ++ met behulp van speciale regels voor codestructurering. De Arduino IDE levert een softwarebibliotheek van het project Wiring, dat veel algemene invoer- en uitvoerprocedures biedt. Door de gebruiker geschreven code vereist slechts twee basisfuncties, voor het starten van de schets en de hoofdprogrammalus, die zijn gecompileerd en gekoppeld aan een program stub `main ()` in een uitvoerbaar cyclisch uitvoerend programma. De Arduino IDE maakt gebruik van het programma AVR om de uitvoerbare code om te zetten in een tekstbestand in hexadecimale codering die in het Arduino-bord wordt geladen door een laderprogramma in de firmware van het bord.



Binnen de IDE spreken we in de taakbalk de tab tools aan en kiezen we daar het bord waarmee we aan de slag gaan. Hierbinnen kunnen we onze sketch schrijven.

De Sketch is een programma geschreven binnen de Arduino IDE. De sketches worden opgeslagen op de computer als een tekst-file met als bestandsexistentie .ino.

Een minimaal Arduino C en C++ programma heeft 2 basisfuncties:

- Setup(): deze functie wordt aangeroepen wanneer de sketch zich opstart of reset. Het wordt gebruikt om variabelen, input en output pinnen en andere bibliotheken die nodig zijn te initialiseren.
- Loop(): na de setup() functie eindigt, wordt de loop() functie uitgevoerd binnen het main programma. Dit bestuurt het bord tot het bord afgesloten of gereset wordt.

Specificaties van de Arduino Uno

Microcontroller	<u>ATmega328P</u>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Tinkerkit

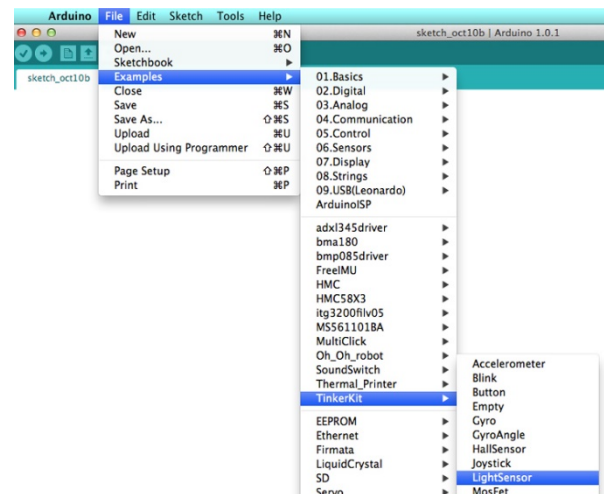
De Uno heeft 6 analoge ingangen, aangeduid met A0 tot en met A5, die elk 10 bits resolutie bieden (dat wil zeggen 1024 verschillende waarden). De Tinkerkit kunnen we verbinden met verschillende input en output devices. Bij deze HAT horen 7 verschillende output devices en 8 verschillende input devices.

Input:

- Button Module x1
- LDR Module x1
- Tilt Module x1
- Therm Module x1
- Rotary Potentiometer Module x1
- Linear Potentiometer Module x1
- Touch Sensor Module x1
- Joystick Module x1

Output:

- Relay Module x1
- 5mm Green Led Module x1
- 5mm Yellow Led Module x1
- 5mm Red Led Module x1
- 10mm Green Led Module x1
- 10mm Yellow Led Module x1
- 10mm Red Led Module x1



Om de tinkerkit aan te spreken hebben we de voorbeeldsketches nodig. Deze waren normaal te vinden op de tinkerkit website, maar die is niet meer beschikbaar. Deze zijn wel nog te vinden op de github repository van tinkerkit. Hier zien we verschillende voorbeeldsketches staan. Voor onderstaande examples hadden we onderandere de LightWithLightSensor sketch nodig.

Deze basissketches zijn makkelijk aan te passen eenmaal we de uitgelezen variabelen hebben en ook de juiste verbindingen verklaren binnen de Arduino IDE.

```
TKLed led (00);
```

Het eerste woord van de regel is de klasse, omdat we een LED aangesloten hebben, het is de klasse TKLed; als we een relais verbinden, zou het TKRelay zijn enzovoort.

Het tweede element is de naam die we toewijzen aan die specifieke LED. Hier gebruiken we gewoon "led", als we met meerdere LED's werken, is het redelijk om ze led1, led2 enzovoort te noemen. Elke component heeft een naam.

Het derde deel is het nummer van de poort waar het blok is verbonden met het schild. Als je het niet zeker weet, kijk dan naar het nummer van de poort waar je kabel in het schild is gestoken.

Alle regels code die we schrijven moeten eindigen met een puntkomma; het is een veel voorkomende fout voor beginners om ze te vergeten, dus maak je geen zorgen als het in het begin niet zo automatisch is.

```
1  #include <TinkerKit.h>
2
3  TKLightSensor lichtSensor(I0); //lichtsensor op poort I0
4
5  TKLed led(00);           //led op poort 00
6
7  void setup() {
8      // initialize serial communications at 9600 bps
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     // waarden van de licht sensor in de variabele brightnessValue opslaan
14     int brightnessValue = lichtSensor.read();
15
16     // veranderd de lichtintensiteit van de led
17     led.brightness(brightnessValue);
18
19     // 10ms wachten voor de volgend loop
20     delay(10);
21 }
```

Examples

Example One Lampje laten branden

We connecteren de lichtsensor met de inputpoort en het LED-lampje met de outputpoort. Hierdoor krijgen we een gesloten circuit.

Om het lampje te doen branden hebben we een bijpassende code nodig.

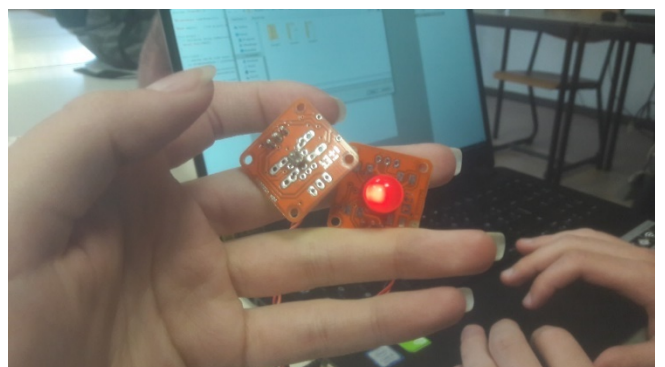
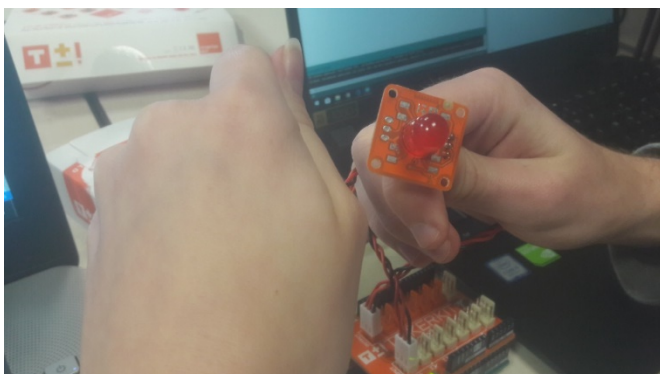
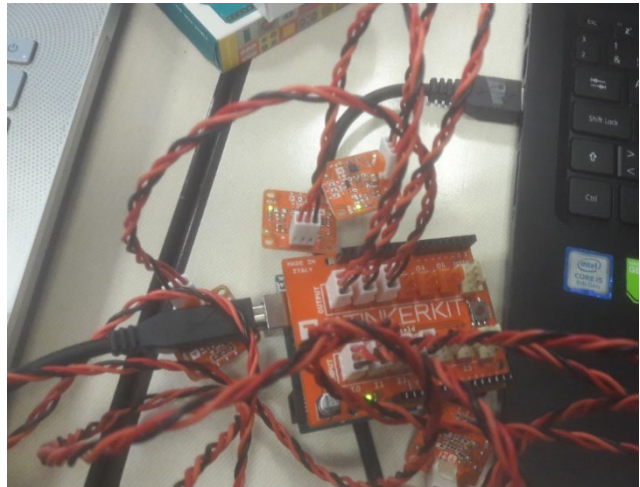
Deze code moeten we schrijven binnen een Arduino-vriendelijke IDE. De software installeren is relatief makkelijk, maar de bijpassende voorbeeld codes zijn iets minder makkelijk te vinden.

De website waarvan we deze moeten halen is niet toegankelijk. Gelukkig is er een github repo met de codes.

Hierbij kiezen we voor de LightWithLightSensor. Deze maakt het mogelijk om de lamp te laten branden.

Het lampje zal nu reageren op het omgevingslicht door deze sensor. We testen dit uit door de sensor te bedekken en daarna weer vrij te geven. Het lampje gaat dimmen zodra er minder licht binnenvalt.

Bij elke verandering aan de code moeten we dit opnieuw doorsturen naar het Arduino-board.



Example Two Lampje dimmen met de potentiometer

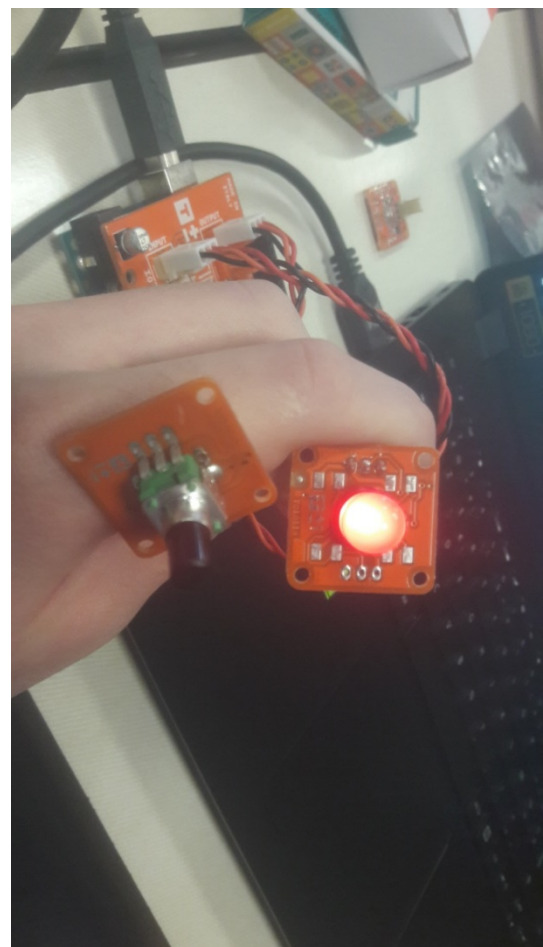
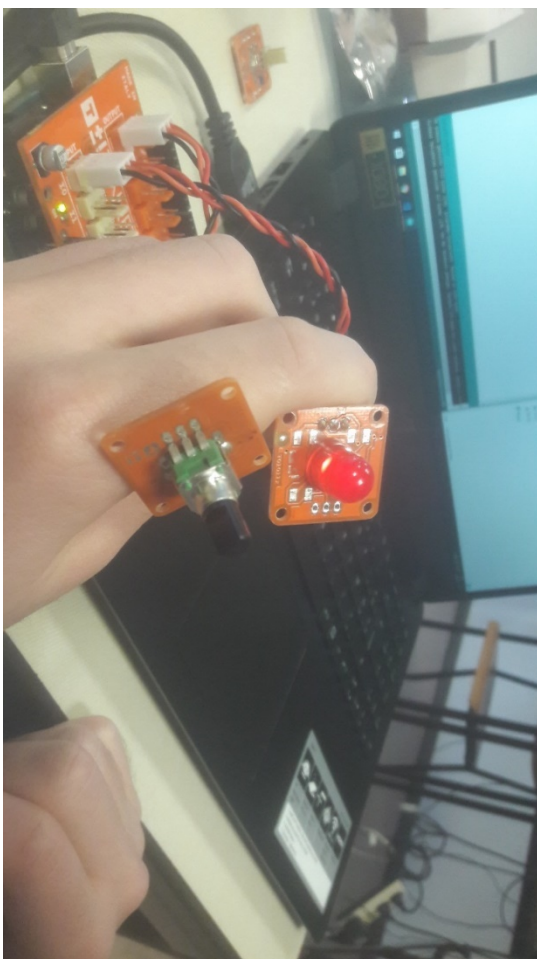
We sluiten de potentiometer aan in plaats van de lichtsensor.

Dit heeft ongeveer het zelfde effect als de lichtsensor, maar in plaats van het werken met de lichtinval werken we met een draaiknop die voor een stabiele dimming zorgt.

Hierbij zien we dat er verschillende waarden zijn die het licht uitstraalt. Deze gaan tot 1023.

Best draaien we de potentiometer volledig open en volledig toe, anders gaan de waarden niet juist zijn in het begin.

Een uitbreiding hiervan is de lineaire potentiometer. Hierbij vallen de waarden tussen 0 en 1020.



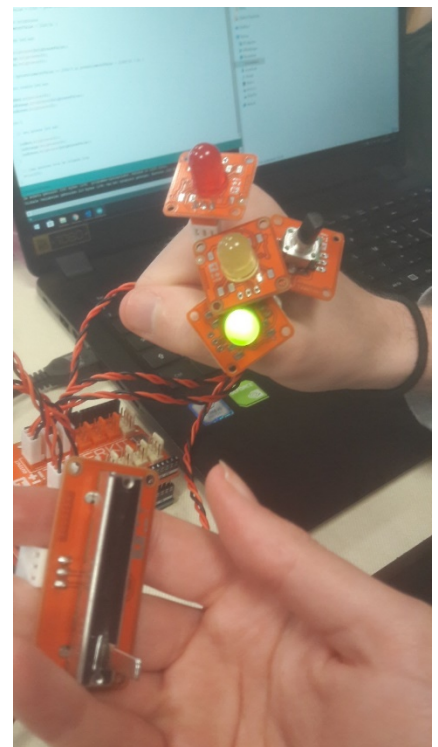
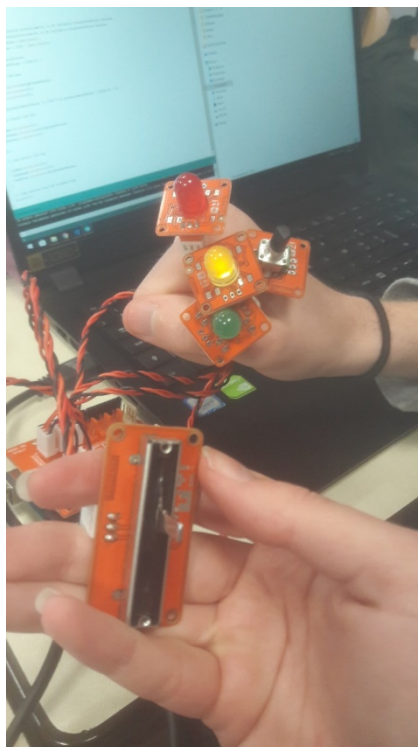
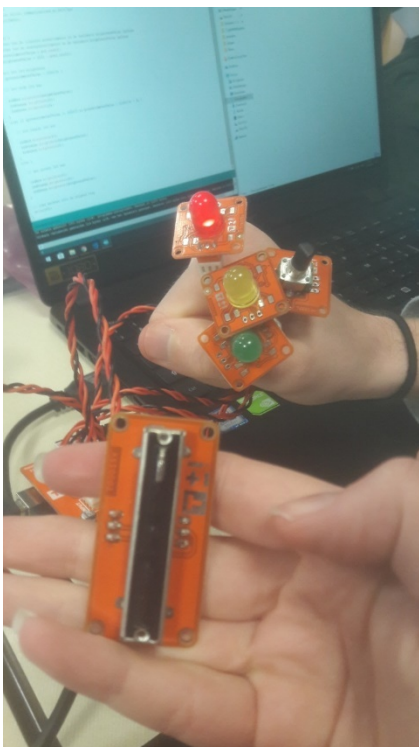
Example Three Het verkeerslicht

In een vorig project probeerden we een verkeerslicht na te maken met de Raspberry Pi. Nu maken we dit met de Arduino.

Hierbij maken we gebruik van de lineaire potentiometer en de 3 grote LED's. Deze plaatsen we in de rood-oranje-groen volgorde.

Elke LED is gekoppeld aan een andere poort. De rode op de 00, de oranje-gele op de 01 en de groene op de 02.

We manipuleren de code zo dat elke licht aanspringt op een bepaalde waarde.



Example Four The Tilt, Touch en Temperature sensor

Elke sensor was via de code aan een andere LED verbonden

The Tilt sensor

Hierbij zien we hoe gevoelig de tilt sensor is aan bewegingen. Bij de lichtste trillingen kan het lampje een onstabiele flikkering geven. Dit is dan vooral als de tilt sensor op zijn zijde ligt.

The Temperature sensor

Op gewone kamer temperatuur blijft het lichtje uit. Maar door het langzaam aan warmer te maken komt er toch en gloed. De warmste temperatuur die we konden genereren was een kleine 32°C. Hiebij zagen we dat het lampje nie zo fel scheen, maar een hogere temperatuur konden we niet bereiken.

The Touch sensor

De touch sensor was vrij gevoelig en reageerde soms met een licht drukverschil dat zich in de buurt afspeelde. Op andere momenten was het de sensor hard vasthouden.

