



Arduino basis elektronica deel 4

Handleiding van Helpmij.nl

Auteur: drejansen

januari 2022

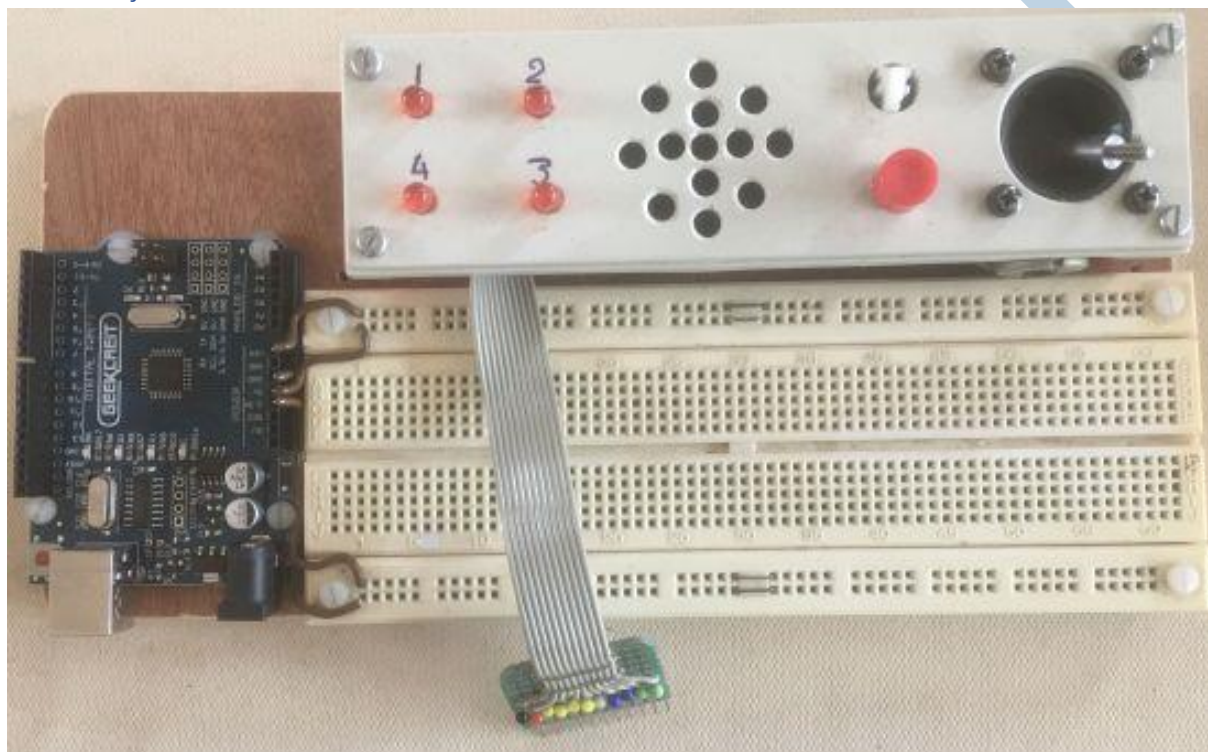
“ Dé grootste en gratis computerhelpdesk van Nederland ”



Cursus

Op het internet staan duizenden Arduino cursussen, ik vind het zinloos om er daar nóg een aan toe te voegen. De meesten zijn in het Engels. Een goede keuze is die van Paul v.d. Veen. Hij legt duidelijk, in het Nederlands, de beginselen van het installeren en programmeren uit. Een PDF bestand kan je [hier](#) downloaden. Er zijn ook Nederlandstalige boeken geschreven, zoals: Arduino voor dummies, waarin onder andere, ook de beginselen van de elektronica wordt uitgelegd. (tot zo ver deze Ster-spot.)

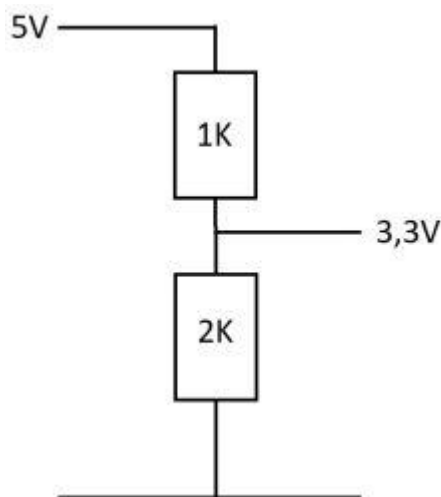
Testbordje



Een opstelling voor je schakelingen, studie, testen en meer. Een breadboardje met Arduino en wat standaard componenten. LEDjes, luidsprekertje, knopjes en een joystick. De LEDs zijn reeds voorzien van een serie weerstand, het luidsprekertje staat in serie met een condensator. De knopjes zijn niet gelijk aan elkaar. Het witte knopje is een standaard knopje, dat dendert bij elke indruk. Het rode knopje is een terugverend wisselschakelaartje, aangesloten op een hardware ontddering zoals in deel -3- is omschreven, met 7400 chip. De joystick bestaat uit twee potmeters, verticaal en horizontaal te bedienen. Eventueel kan je de schakeling nog uitbreiden met een 2x16 LCdisplay, al dan niet in I2C uitvoering. Uiteraard kan je hier zelf oneindig mee knutselen. Ik heb het volgens de foto gemaakt en op een plankje gemonteerd. Het breadboard heeft 4 horizontale voedingslijnen, die ik als volgt heb toegewezen. De bovenste lijn is de aangelegde spanning meestal 12V of 9V, uit de externe voeding. Daaronder de 5V uit de Arduino zelf. De derde lijn, de op een na onderste, is aangesloten op de interne 3,3V van de Arduino, helemaal onderaan de ground. Uiteraard mag je hier zelf anders over beslissen, maar zo heb ik het gedaan. De horizontale lijnen hebben halverwege een onderbreking, vergeet niet om door te verbinden. Het 'stekertje' is gemaakt van knopspelden uit moeders naaikistje. Gesoldeerd op een klein stukje gaatjes print. De onderlinge afstand van de naalden past dan precies in de gaatjes van het breadboard. Achtereenvolgens: zwart = GND; rood = +5V; vier keer geel, de vier ledjes; wit de luidspreker; twee blauwe de knopjes; twee groene de twee potmeters van de joystick.

Een en nul

Ja, dat weten we nu wel: 0V is -0- en 5V is -1-, maar hoe zit het dan met 0,5V en 4V? En wat als de spanning nog wat lager/hoger is? Want wanneer de draadjes wat langer worden, en/of er wat meer belasting aan hangt, komen praktijk en theorie niet meer overeen met elkaar. Alles kleiner dan 1V is -0- alles groter dan 2,5V is -1-. Wat dan als het 2V is? Welnu, dan weet Arduino het ook niet. Dat moet je niet willen, dus zorg er voor dat de signalen het juiste niveau hebben! Er zijn sensoren en actuatoren die op 3,3V werken, welnu, daar heeft Arduino geen moeite mee, immers 3,3V ligt duidelijk boven 2,5V. Arduino zelf geeft 5V af, daar is niet elk device van gecharmeerd. Een bluetooth module wil 3,3V.



We gaan even terug naar deel -1- van deze reeks: serie schakelen van weerstanden. Wanneer je 1K en 2K in serie schakelt: $5V \times 2 / (1+2) = 3,3V$. Maar 2K is er niet, wel 2K2. dan wordt het 3,4V, geen probleem. Weerstanden zijn er niet in elke waarde, de meest gebruikte waarden zitten in de E-12 reeks: 10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82. Er zijn meerdere reeksen, de 2K weerstand komt wél voor in de E-24 reeks: 5, 24, 10, 11, 12, 13, 15, 16, 18, **20**, 22, 24, 27, 30, 33, 36, 39, 43, 47, 51, 56, 62, 68, 75, 82, 91. Niet elke 'vonkenboer' heeft deze weerstanden in huis. Mocht je überhaupt nog een elektronica onderdelen zaak weten te vinden :-). Reken zelf uit hoe de spanning wordt bij 1K2 en 2K2 weerstanden, die wel in de E12 reeks zitten. Er zijn chipjes die heen en weer werken, enerzijds 3,3V, anderzijds 5V. Vier van die convertors op een rij koop je hier: [Level convertor](#). Nog even naar de bluetooth convertor: pin -0- RX en pin -1- TX. RX moet naar TX van de bluetooth. Pin -1- TX moet via de **weerstandsschakeling** naar RX van de bluetooth. Wanneer je pinnen 0 en 1 gebruikt, werkt de monitor en upload niet meer. Beter is het om twee andere pinnen te gebruiken en een software seriewaarde te programmeren. Hoe dat moet, staat allemaal in de cursus van Paul. (en vele anderen.)

I2C en SPI

Arduino heeft meerdere communicatie mogelijkheden: I2C. De pootjes A4 = SDA en A5 = SCL. (SDA = Serial Data en SCL = Serial Clock.) Dit is een communicatielijn waarop je meerdere apparaten kan aansluiten, zowel sensoren als actuatoren! Ook poort uitbreidingen en geheugen uitbreiding zijn probleemloos. Nadeel is de traagheid, maar dat hoeft niet altijd een probleem te zijn. (Hoe en wat? Zie Arduino cursus!)

Dan is er de SPI poort, waarmee je buiten de bootloader om in de Arduino kan komen. (SPI = Serial Peripheral Interface) Dit is iets voor gevorderden, het zijn de 6 pinnetjes op het bordje. 1 MISO; 2 VCC; 3 SCK; 4 MOSI; 5 reset; 6 GND. Het zijn de pootjes op het bord: D10, D11, D12 en D13 vormen de SPI connector. (MISO = Master In Slave Out; MOSI = Master Out Slave In; GND = ground; VCC =

+5V). **Let op!!** Er zijn twee SPI connectors, vlak bij de USB connector zit er óók een, die is voor de USB processor. **AFBLIJVEN!!!**

Voedingsprobleem

'Onzichtbaar' voedingsprobleem: Veronderstel dat je een motor of andere wat grotere belasting hebt, die je via Arduino wilt aansturen. Je gebruikt de 5V uit de Arduino, immers servo motoren werken op 5V, wel zo handig. De aanloopstroom van een motor is relatief groot in vergelijking met de nominale motorstroom. Als meerdere servo's tegelijkertijd starten, kan het zijn dat door die aanloopstroom, de voedingsspanning heel even te veel daalt. Hierdoor zal de Arduino resetten. Op het grensgebied, soms wel, soms niet. Dat is heel vervelend, zoeken maar! Schakeling in orde, software foutloos geschreven en toch die rare 'af-en-toe' storing. Motoren en andere grotere belastingen kan je daarom het best op een afzonderlijke voeding aansluiten. Uiteraard moet je de massadraden van al die voedingen met elkaar verbinden. Probleem opgelost!

Sneltoetsen

Handig om te weten, werkt een stuk sneller dan via de taakbalk.

Ctrl+R Verifiëren en compileren
 Ctrl+U Uploaden
 Ctrl+Shift+M Seriële monitor
 Ctrl+T Automatische opmaak van je code
 Ctrl+, Voorkeuren openen

Tekst opslaan in Arduino kleuren

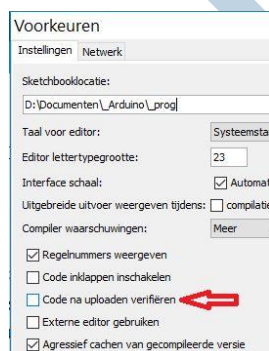
Ik maak soms listings van programma's, maar dan liever in de Arduino kleuren. Dat doe je als volgt: In menu bewerken >> opslaan als HTML >> naar: kladblok bestand.html (Verander de extensie TXT in HTML)

Dit bestand met de browser openen (In de browser zie je kleur!) >> de inhoud kopiëren naar het document.

Sneller uploaden

Door in instellingen >> voorkeuren >> het vinkje bij 'code na uploaden controleren' weg te halen, werkt het uploaden veel sneller. Verifiëren kan je ook tijdens het programmeren doen.

Hier kan je ook de regelnummers in/uit schakelen, ook erg handig



Het geheugen

De Arduino heeft 2K RAM geheugen en 32K flash geheugen (= EEPROM). Het programma komt in het flash geheugen terecht, de variabelen in het RAM geheugen. Het flashgeheugen blijft bestaan als de boel wordt uitgeschakeld. De inhoud van het RAM geheugen gaat verloren na uitschakelen. Een opdracht als: `Serial.println("Hello world!");` kost 12 RAM bytes. Zijn er meerdere van dit soort opdrachten, dan raakt je RAM snel vol. Dit soort meldingen kunnen beter in flash geheugen worden opgeslagen, dat doe je door: `Serial.println(F("Hello world!"));` De `F(...)` functie geeft aan de compiler

de opdracht om deze tekst in flash geheugen te zetten.

"Extra" digitale pootjes

Er zijn nog meer digitale pootjes, namelijk 14 t&m 19.

De analoge pootjes kunnen als digitale pootjes worden ingezet: A0 = D14 en A1 = D15 etc.

Voedingskeuze:

USB of Extern (Vin)? Als beiden zijn aangesloten heeft Vin (extern) de voorkeur!
De communicatie via de USB-poort blijft uiteraard intact.

Referentie spanning

Standaard is de voedingsspanning de referentie spanning. Wil je dat anders, dan doe je dat als volgt:

Voedingsspanning: `analogReference(DEFAULT);`

Interne spanning 1,1V: `analogReference(INTERNAL);`

Extern pootje: AREF: `analogReference(EXTERNAL);` sluit hier jouw eigen Uref op aan. max 5V=

Interne timer

Er is een altijd lopende timer: millis, de hartslag van Arduino. 'Altijd' is toch eindig, na 50 dagen begint de teller opnieuw te tellen. Dat zou fouten kunnen geven. De hartslag begint te lopen zodra je hem aan zet. na 50 dagen zal de long-variabele millis overlopen en -0- worden. Maar binnen die tijd hebt je hem vast wel een keer uitgeschakeld. Theoretisch maximum 4.294.967.295 (miliseconden = ca 50 dagen).

Pull-up en pull-down

Er is een interne pull-up weerstand, dus die hoeft je niet extern toe te voegen. Een pulldown weerstand is niet nodig, want de aangesloten sensor heeft een niveau tov massa. Het grond niveau is daarmee vastgelegd. Schakel knopjes en LEDjes altijd naar massa, gebruik voor knopjes de interne pull-up weerstand. In de cursus staat hoe je die aan/uit zet.

Uitgangsstroom, ingangsstroom

Je kan een Led 'aan' zetten door de uitgang hoog te maken. De andere led-zijde is dan (via een weerstand) op de massa aangesloten. Je kan een led ook aan zetten door de uitgang laag te maken, hierbij is de andere ledpoot op de + aangesloten. Wat is het verschil? Een Arduino uitgang is een push-pull schakeling, totempole of hoe je die ook wilt noemen. Dat zijn twee transistoren waarbij de collector van de bovenste tor via een weerstand met de + is verbonden, de emitter van de onderste tor ligt aan massa. Het 'pootje' zit op het midden, aan de emitter van de bovenste en de collector van de onderste tor. (de werkelijkheid is complexer, maar voor de uitleg volstaat dit verhaal.) Stuur je de LED met 'HIGH' aan, dan loopt de ledstroom ook door de weerstand van de bovenste tor. Stuur je de led met 'LOW' aan, dan blijft deze interne weerstand stroomloos. Minder spanningsverlies maar ook minder interne warmte ontwikkeling in de Arduino processorchip. Dus naar massa schakelen heeft mijn voorkeur.