



MAASPOORT
KIDS

**BOUW JE EIGEN
WEKKER**

Handleiding

mail: activiteiten@maaspoortkids.nl

In samenwerking met CoderDojo Maaspoort



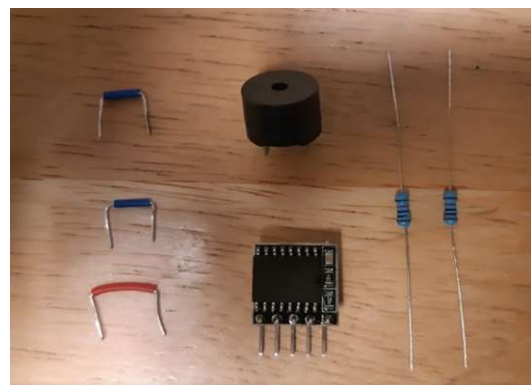
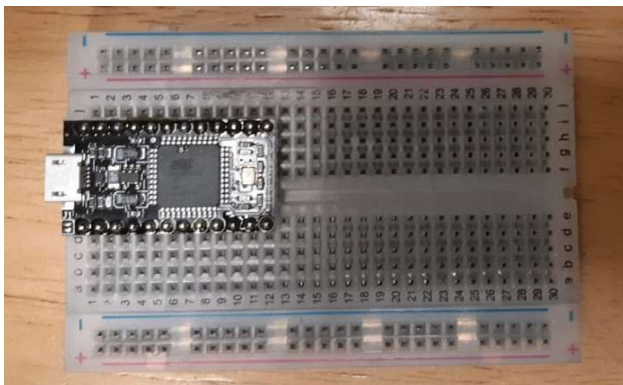
Vooraf

Deze handleiding beschrijft hoe je zelf een wekkertje kunt bouwen op basis van de door Maaspoort Kids en CoderDojo Maaspoort samengestelde bouw pakketjes. Deze pakketjes bevatten wat je nodig hebt om het binnenste van het wekkertje te bouwen. Daarnaast heb je nog de buitenkant nodig en daarvoor kun je gebruiken wat je wil. Hiernaast zie je bijvoorbeeld een wekkertje dat in een juwelenkistje gebouwd is.



De onderdelen

Hieronder zie je een overzicht van wat in het pakketje zit. Wat de onderdelen doen wordt in de rest van de beschrijving uitgelegd.



Arduino's

Arduino's zijn een soort kleine mini-computertjes ('controllers') waarvan je makkelijk kunt regelen hoeveel stroom er op de pinnetjes staat. Daarmee kun je allerlei andere bouwsteentjes zoals beeldschermpjes, knopjes, maar ook diverse andere dingen aansturen.



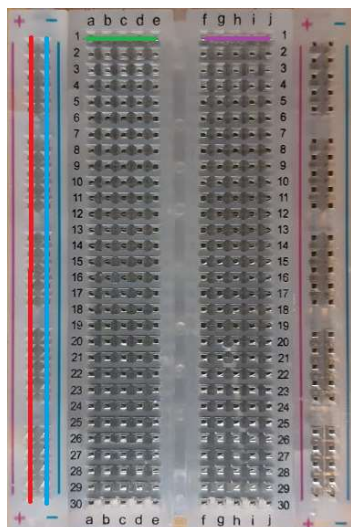
Het programmeren laten we in eerste instantie buiten beschouwing. De arduino in het bouw pakket is al voorgeprogrammeerd, maar als je interesse hebt kunt je de software van onze github pagina (<https://github.com/maaspoortkids/wekker>) halen en zodanig aanpassen dat je zelf je eigen welkomstboodschap erin kunt programmeren. Het uitleggen hiervan gaat wat te ver voor deze handleiding en het is natuurlijk ook niet noodzakelijk want het staat

er al op. Mocht je er interesse in hebben kom je met wat rondgooglen al een heel eind. Als je hiermee aan de slag gaat is het goed om te weten dat het type in het pakketje een 'Pro Micro' heet die op 5 volt werkt.

Breadboards

De basis waar we mee starten heet een breadboard. Dit is een plaatje met allerlei gaatjes erin waar je makkelijk elektrische componenten in kunt klikken.

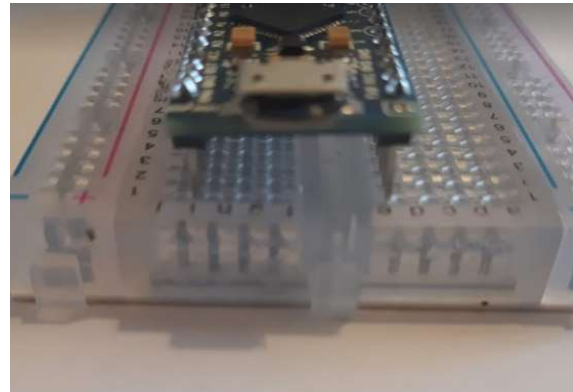
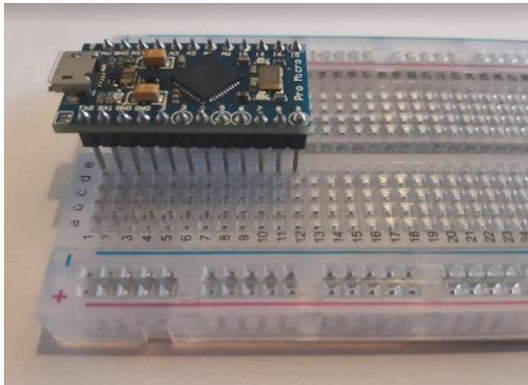
Binnenin zijn ze als volgt verbonden. De blauwe en rode lijnen in de lengterichting worden de 'voedingsrails' genoemd. Blauw is - en rood is +. Andere benamingen zijn 'GND' (van 'Ground', ofwel -) en 'VCC' (+). Daarnaast zijn de rijen van a t/m e en van f t/m j verbonden. E en f maken onderling dus geen contact.



In het algemeen is het handig de linker en rechter VCC en GND rails met elkaar te verbinden (VCC met VCC en GND met GND), maar voor onze wekker is dat verder niet nodig en doen we dit niet.

Deze dingen heten breadboard omdat hier vroeger echt houten broodplanken voor gebruikt werden. Met spijkers en draad kon je dan dingen aan elkaar knopen, en zoals je ziet werkt de moderne manier toch wat makkelijker.

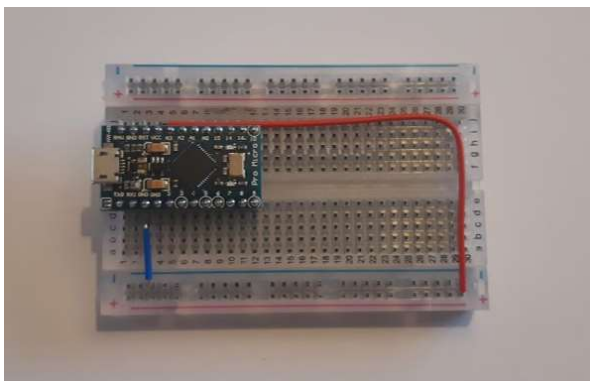
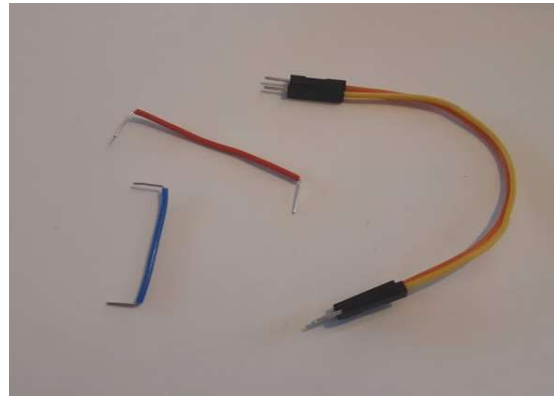
Druk de arduino in het breadboard op de plaats zoals op de foto's getoond. Op de foto's staat hij nog bovenop het bord maar het is de bedoeling hem er voorzichtig in te drukken. Als het niet makkelijk gaat is het beter een beetje heen en weer te bewegen in plaats van hard te duwen. De linker pin zit op positie e1.



De draadjes

Met deze breadboards gebruikt je zgn 'dupont jumper draadjes' om verbindingen te leggen. Als alternatief kun je ook dunne, stugge draad gebruiken. In deze kit zitten beide. De stugge draad gebruiken we voor de + en - verbindingen en de jumper draadjes voor de andere signalen.

Technisch werken ze hetzelfde. Door de stugge draad voor makkelijke routes te gebruiken blijft het geheel wat overzichtelijker. De jumperdraadjes koop je in allerlei kleuren waarbij je de kleur niet kunt kiezen. De kleur betekent dus in tegenstelling tot wat je bij dingen als stopcontacten ziet dus ook niets. Hooguit is het een goed gebruik om voor de - zoveel mogelijk blauw of zwart en voor de + rood te gebruiken.



Neem het lange rode draadje en een kort blauw en sluit deze aan zoals hieronder. De 'VCC' (= +), breadboard gat j4 van de arduino naar de rode VCC rail onderin. De blauwe moet van de GND (= Ground, ofwel de -) naar de blauwe GND rail onderin. Deze kun je vastzetten in het breadboard gat c3 of d4 afhankelijk van met de lengte het mooist uitkomt.

De knopjes

In het pakket zitten twee knopjes waar de draadjes al aan vast gesoldeerd zijn. Je hoeft deze nog niet aan te sluiten.



Pull down weerstandjes

De knopjes sluiten we nog even niet aan omdat die draden een beetje in de weg zitten, maar de voorbereiding maken we wel al. Hiervoor heb je weerstandjes nodig. De werking van zgn. pull down weerstandjes is wat lastig uit te leggen. De samenvatting: je hebt ze gewoon nodig.

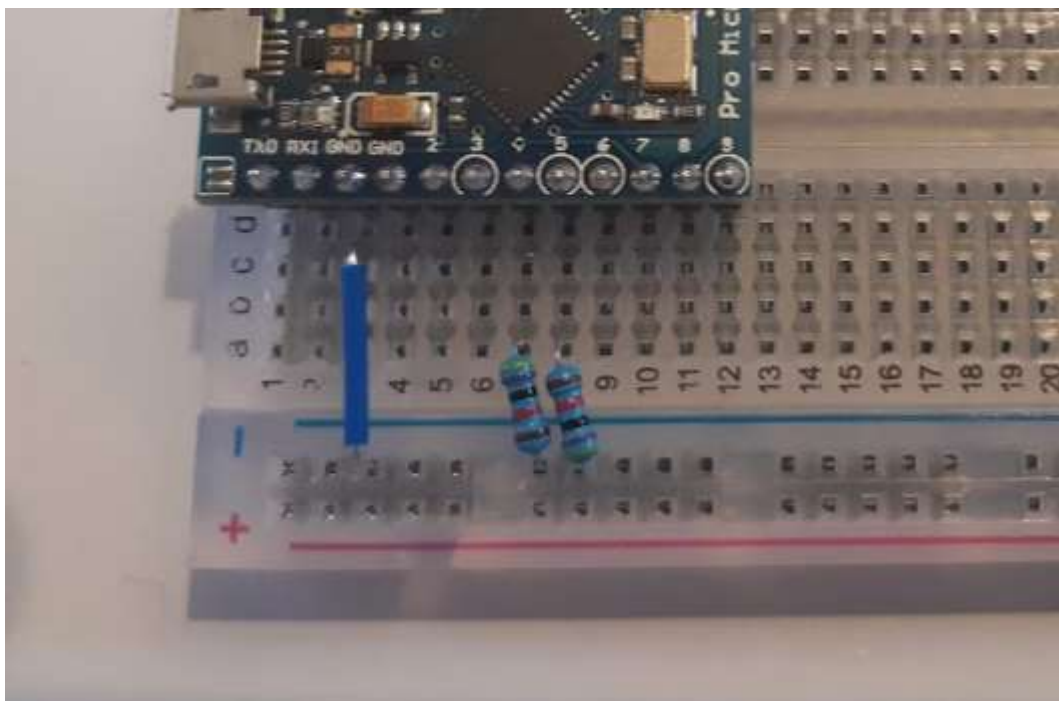
Voor degenen die toch de achtergrond willen weten volgt hier een vereenvoudigde uitleg.

Als je een tuinslang hebt je duwt er met je voet op komt er minder water uit. Hoe harder je trapt hoe dunner de doorgang wordt en hoe minder water er doorheen komt. Weerstandjes doen hetzelfde met stroom en dit wordt uitgedrukt in een aantal 'Ohm'. Dit is de eenheid die naar de uitvinder hiervan vernoemd is. De grootte van het getal geeft daarbij aan hoeveel *tegengehouden* wordt en dus niet hoeveel stroom er nog doorheen komt, dus bij een heel hoge Ohm-waarde komt er nog maar weinig stroom doorheen.

De draadjes die aan de knoppen zitten verbinden als de knop ingedrukt wordt de + en de arduino inputpin waar ze aan vast zitten. De arduino checkt dus de hele tijd of er via de knop stroom op de pin staat. Als de knop niet ingedrukt wordt is dat niet zo en als hij wel ingedrukt wordt is dat wel zo.

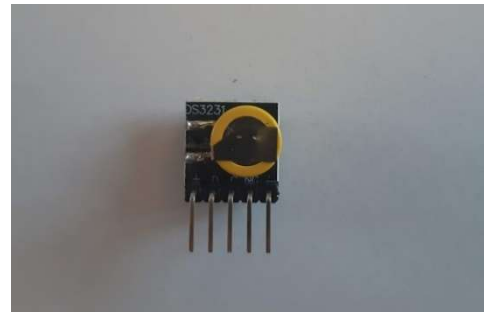
Tot zover is het simpel. Waar het ingewikkeld wordt is dat er als er geen stroom op staan de pinnetjes van zichzelf ook wat ruis veroorzaken. De programmacode ziet dit verschil niet. Om van deze ruis af te komen kun je een draadje naar de GND trekken om deze ruis-stroom af te voeren. Daarbij moeten we wel voorkomen dat alle stroom die bij het indrukken van de knop naar de pin moet gaan wegstroomt. Dit doen we door niet zomaar een draadje naar de GND te trekken maar door hiervoor een weerstandje te gebruiken. Als vuistregel is een weerstandje met een waarde van 10k (=10.000) Ohm een goede keuze.

Knip de weerstandjes op de juiste lengte af (ongeveer hetzelfde als het korte blauwe draadje tussen GND en de blauwe balk) en bevestig ze tussen de blauwe GND balk en a7 en de andere tussen GND en a8. Op deze manier zorgen ze dat we de arduino pinnen 4 en 5 later voor de knoppen kunnen gebruiken.



Het klokje

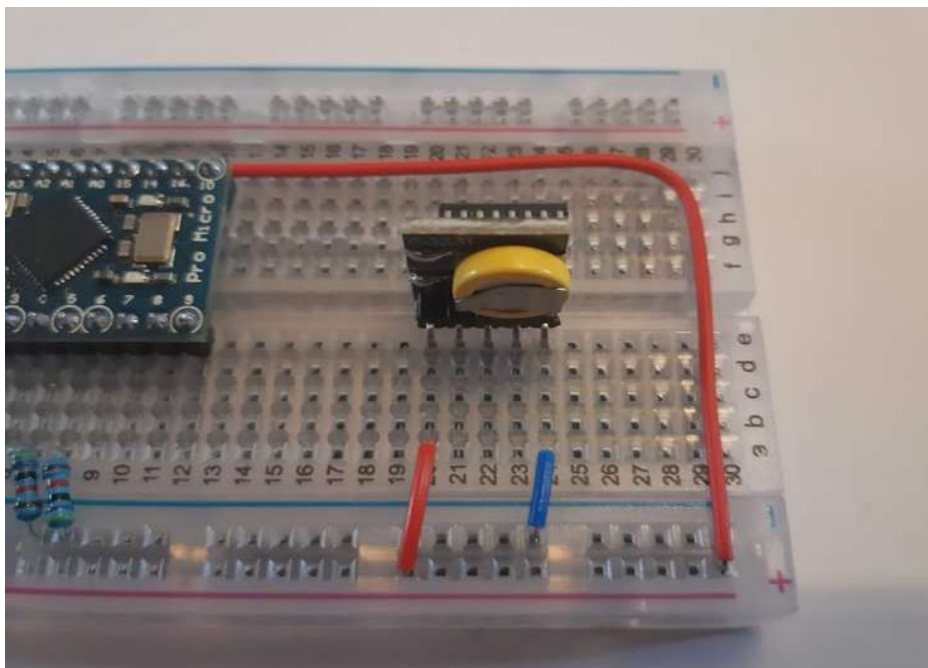
De tijd wordt bijgehouden in het kleine losse printje. Hierop zit een chip waar de klok zelf in zit en een batterijtje om ervoor te zorgen dat de tijd ook onthouden wordt als de wekker niet ingestoken is.



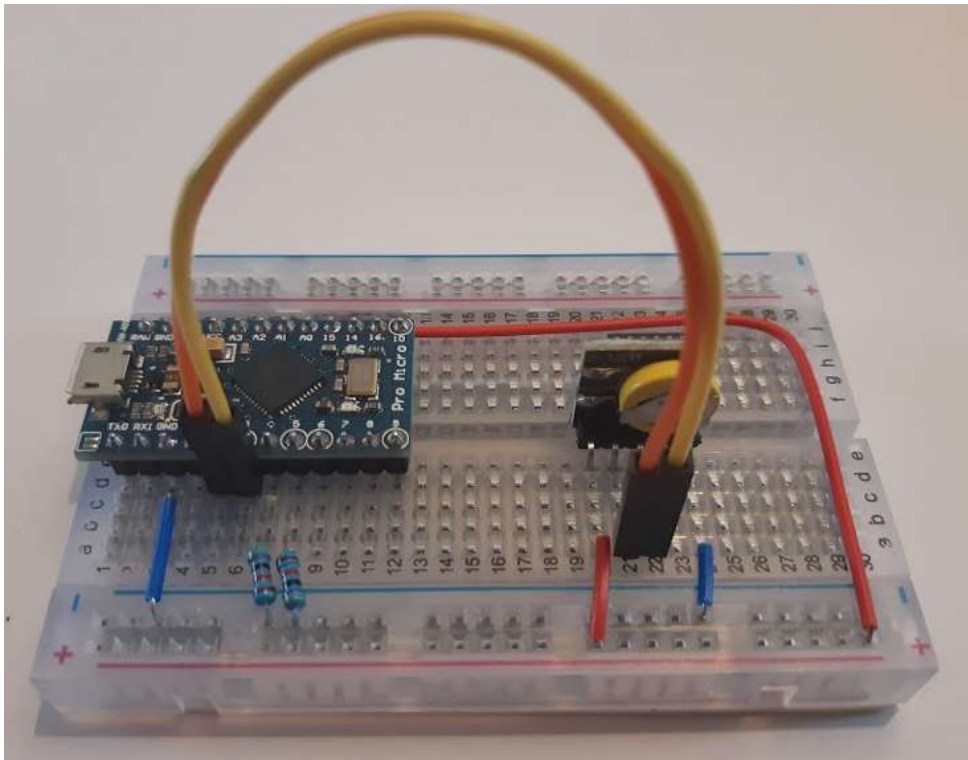
I2C

Vroeger waren alle dingen zoals displays, klokmodules en andere sensortjes verbonden met hun eigen draadjes. Maar dan heb je al snel heel erg veel draadjes nodig, en dus hebben mensen iets bedacht om meerdere dingen via de zelfde draadjes te laten praten. Hier zijn verschillende manieren voor. Een ervan heet 'I2C' en dat is de manier hoe het klokje en ook het display met de arduino praten. I2C apparaten gebruiken een + en - voor de stroom. Daarnaast gebruiken ze 2 draadjes om hun signalen heen en weer te sturen. In het geval van het klokje is dat dus de tijd en voor het display is dat welke tekst er getoond moet worden. Het klokje heeft daarnaast nog een vijfde pinnetje dat niet gebruikt wordt.

We gaan eerst die twee stroomdraden trekken. Kies hiervoor een kort rood en blauw draadje en sluit ze aan vanaf de VCC en GND rail aan de rand van het breadboard. Zorg dat je 3 pinnetjes tussenruimte hebt. Het maakt verder niet uit waar je ze plaatst. Aansluitend plaats je het klokje. Let erop dat de + en - tekenjes op het klokprintje overeenkomen met de rode (+) en blauwe (-) draad. Technisch maakt het niet uit of je ze in rij e of bijvoorbeeld rij b zet, maar om het makkelijk passend te laten zijn kun je het beste rij e gebruiken.

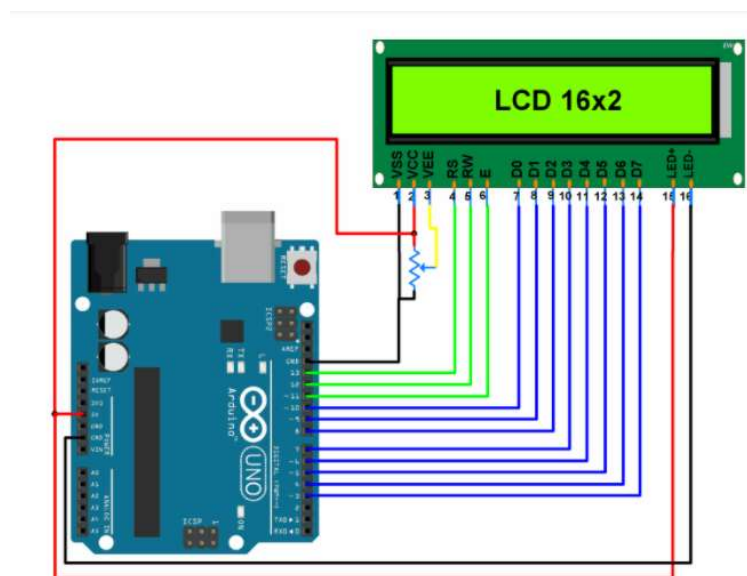


De ongebruikte pin zit in deze foto op rij 23. Vervolgens sluiten we de signaalkabeltjes aan. De I2C pinnen op de arduino zijn pin 2/rij 5 en 3/rij 6. Neem 2 jumperkabels en sluit pin 2 aan op rij 21 en pin 3 op rij 22.



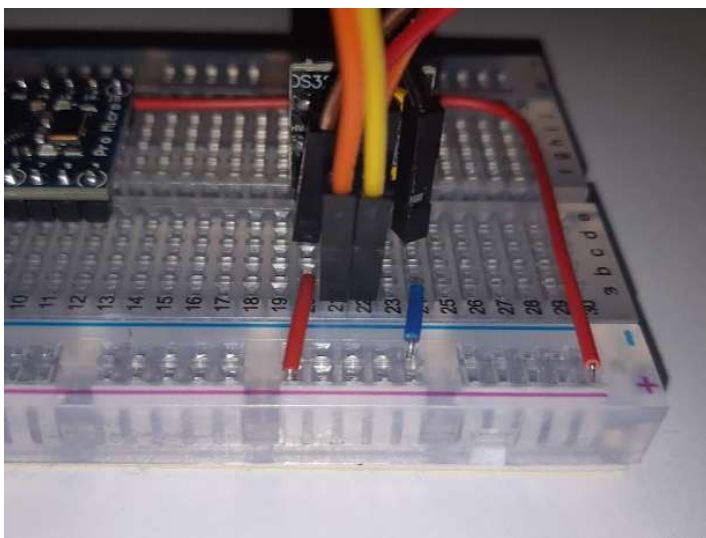
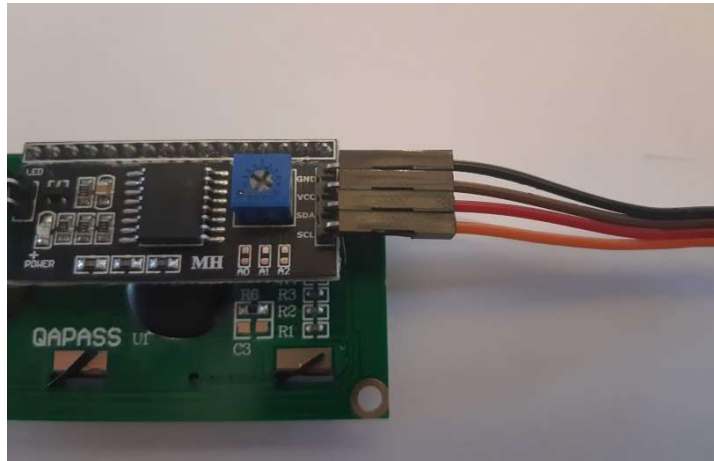
Het display

Het display komt ook op de I2C lijnen. Dit display heeft een losse I2C adapter. Dit is het kleine printje dat achterop gesoldeerd is en hier zie je ook gelijk waarom dit handig is. Hieronder krijg je een idee hoeveel draden je nodig zou hebben zonder dit losse printje. Gelukkig hoeft dat dus niet.



Dit printje vertaalt deze hele rij signalen dus naar handige gegevenspakketjes die met een stuk minder pinnetjes verzonden kunnen worden.

Zet 4 jumperkabeltjes op de pinnetjes van het display.



Sluit deze vervolgens op het breadboard aan. GND pin aan de blauwe GND rail op het breadboard, de VCC pin aan de rode VCC rail, SDA op rij 21 en SCL op rij 22. Rij 23 blijft dus leeg. Zoals gezegd betekenen de kleuren verder niets en op de foto's zie je dan ook dat voor signaalkabel en displaykabels oranje op een andere manier gebruikt wordt. Zoek hier dus geen logica in want die is er niet.

Voor de duidelijkheid hier nog een foto zonder de kabels die naar de arduino gaan.



Helderheid en contrast van het display

Soms is het display wat fel. In dat geval kun je het jumperbruggetje dat de twee pinnetjes verbindt vervangen door het jumperdraadje waar een weerstandje tussen gesoldeerd zit. Het meegeleverde weerstandje is 4k7 (=4.700) Ohm maar mocht dat te fel of te donker zijn kun je dit eenvoudig door een andere waarde vervangen.



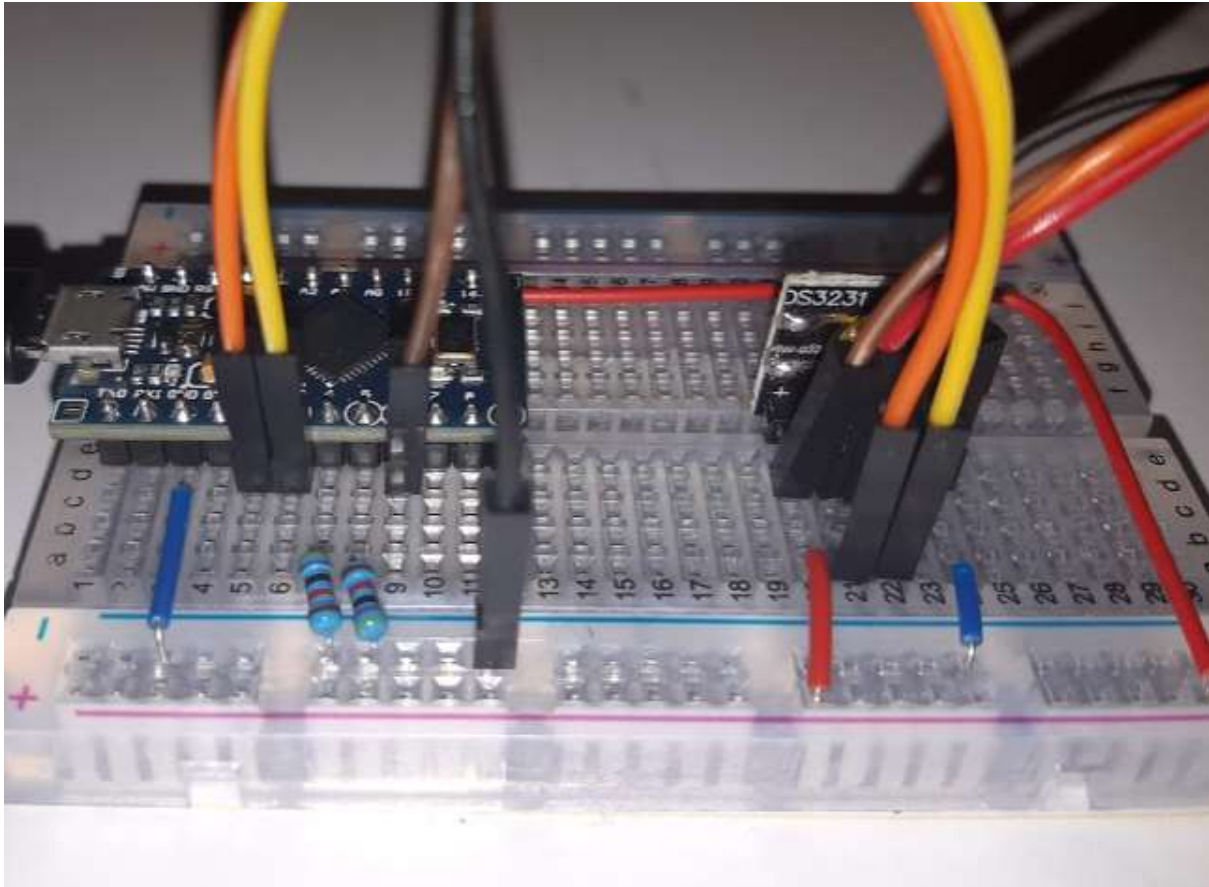
Het contrast kun je bijstellen met het blauwgrijze stelschroefje rechts op de foto.

De zoemer

De zoemer is een simpel apparaatje. Koppel twee jumperkabeltjes aan de zoemer. Let hierbij op de + markering op het zwarte tonnetje.

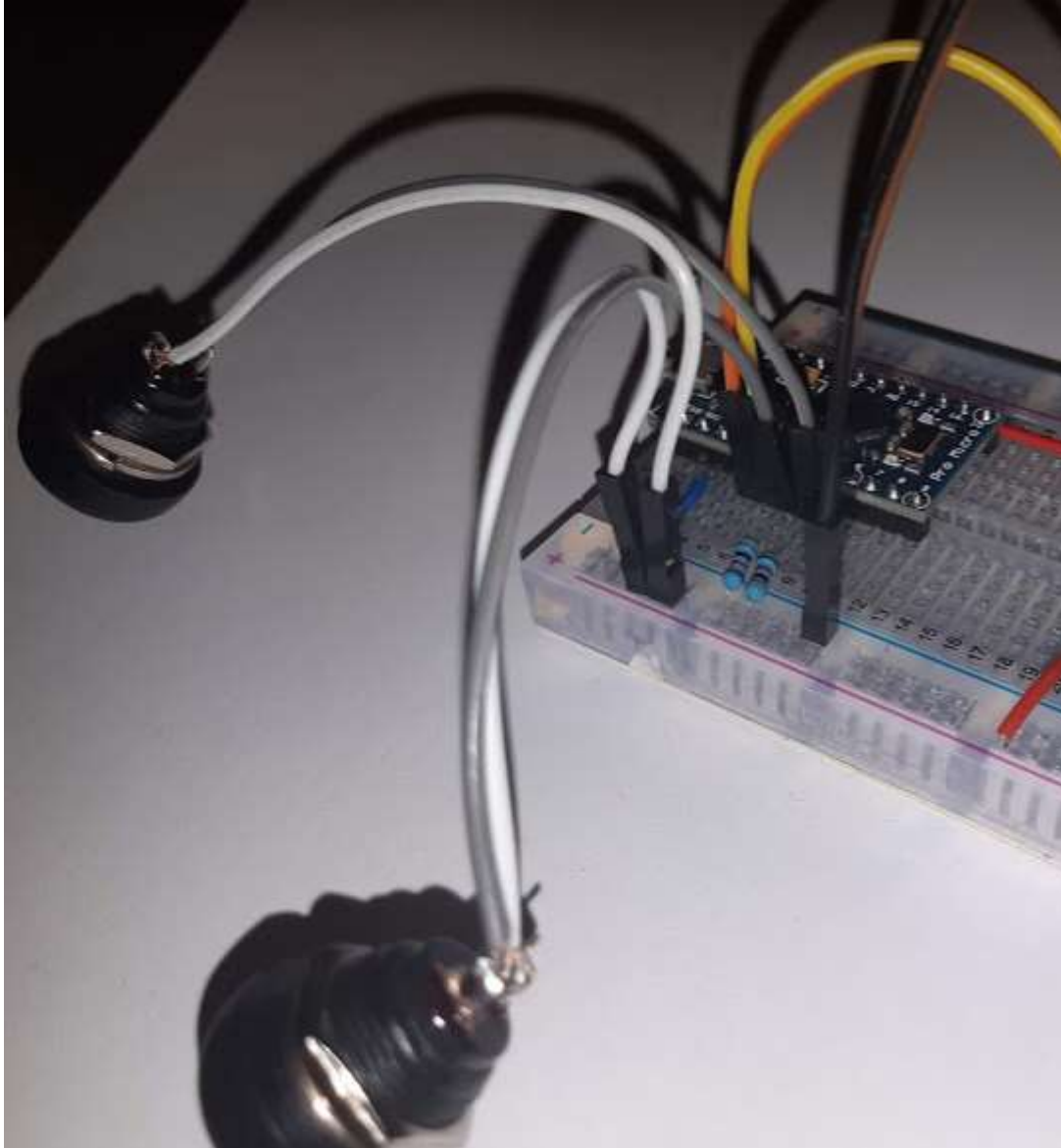


Deze + sluit je aan op pin 6/rij 9 en de – op de blauwe GND rail.



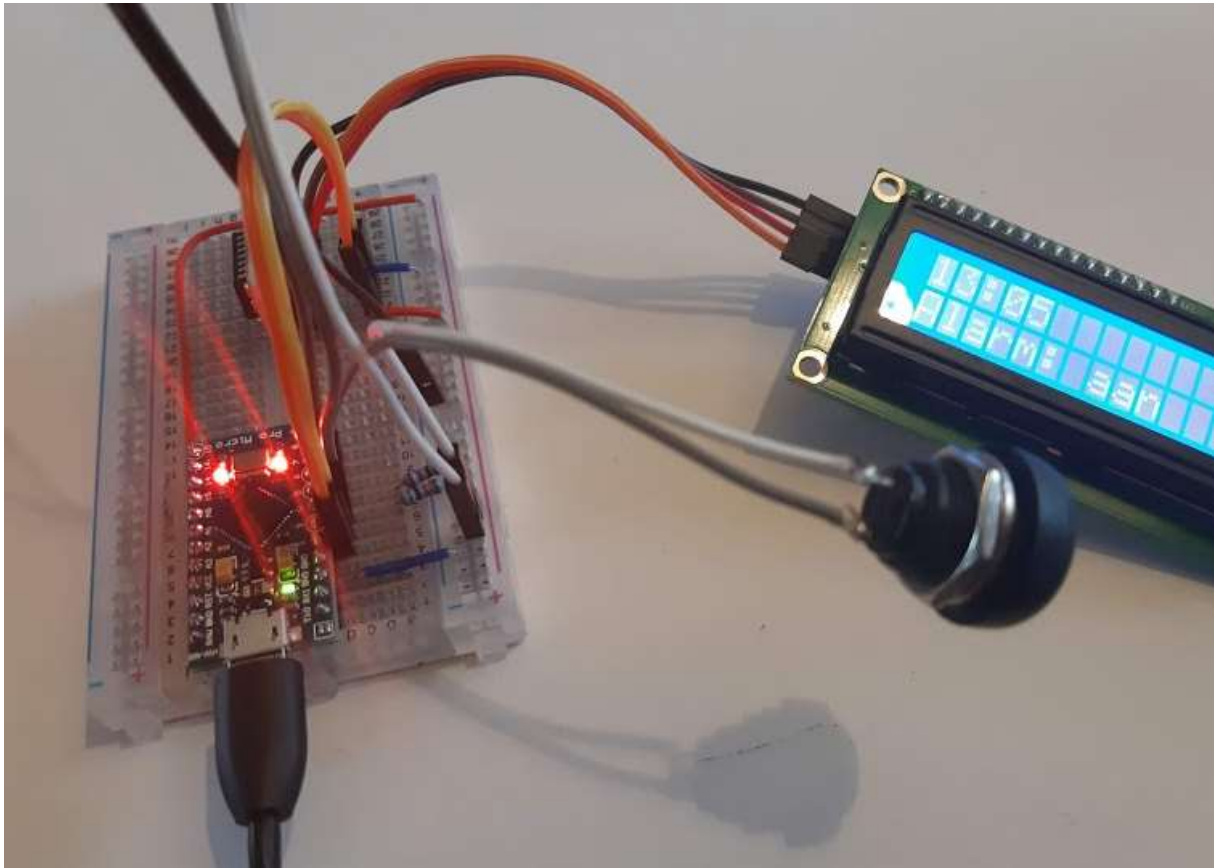
De knoppen

Tenslotte sluiten we de knoppen aan. Deze sluit je aan op de pinnen 4 en 5 waar ook al de pull down weerstandjes zitten. Ze moeten tussen arduino en weerstand aangesloten worden dus het beste is om ze in de rij direct tegen de arduino aan te prikken. Van elke knop gaat het andere draadje in de rode VCC rail. Knoppen hebben geen + of -, dus het maakt niet uit welk draadje je waarvoor gebruikt.



Het resultaat

Als je alles goed gedaan hebt en je de USB kabel insteekt heb je nu een werkend klokje!



De behuizing

Tja, hier begint je eigen fantasie. Dit deel valt buiten het bouw pakket. Zoals op de eerste foto getoond is heeft m'n dochter hiervoor haar juwelenkistje omgebouwd, maar er is natuurlijk van alles denkbaar. We zijn heel benieuwd naar wat jullie ons gaan mailen op activiteiten@maaspoortkids.nl want we willen natuurlijk wel het resultaat zien :)

De stroomkabel

Je kunt de arduino met een adapter en een losse micro-usb kabel aansluiten. Als je gewoon een goede adapter gebruikt is dit helemaal veilig, maar het is wel belangrijk om te zorgen dat het printplaatje heel blijft als er per ongeluk aan de kabel getrokken wordt. Als je dit niet zou doen zou waarschijnlijk de connector van de printplaat scheuren. Je kunt dit voorkomen door een knoop in de kabel te leggen. Als het gat wat groot is of de kabel vrij dun kun je nog een dop van een stift door de knoop heen. Als er nu eens aan de kabel getrokken zou worden wordt de dop tegen het hout aangetrokken en komt er geen kracht op de arduino printplaat te staan.

Om kortsluiting te voorkomen moet je in ieder geval nooit iets van metaal gebruiken.



Als je deze nu in de arduino steekt is je wekker klaar voor gebruik.

De gebruiksaanwijzing

Je wekker heeft 2 knoppen: Knop 1 om langs de verschillende instellingen te kunnen lopen en knop 2 om de instelling te veranderen.

Dus:

Je ziet de tijd > knop 1 > je kunt de uren instellen met knop 2 > knop 1 > je kunt de minuten instellen met knop 2 > knop 1 > je kunt de uren van het alarm instellen met knop 2 > knop 1 > je kunt de minuten van het alarm instellen met knop 2 > knop 1 > je ziet de tijd weer.

En: je ziet de tijd > knop 2 > het alarm gaat aan of uit

Oplossen van problemen

De standaardoplossing als de wekker niet werkt is om eerst alle draadjes te controleren. In het geval je alleen vierkante blokjes op het scherm zit zijn er twee dingen die je kunt doen.

Als eerste wissel je de signaalkabels van het display om (rij 21 en 22) en reset je de wekker door de USB kabel uit te trekken en opnieuw in te steken. Als dat niet helpt zet je ze weer terug zoals ze zaten, reset je hem weer en kun je het contrast bijstellen door aan de blauw/grijze stelschroef op het display te draaien

Over CoderDojo

Deze workshop wordt georganiseerd door Maaspoort Kids in samenwerking met CoderDojo Maaspoort. CoderDojo heeft onder andere meegeholpen door de software die erop staat te schrijven. Maar wat is CoderDojo nu eigenlijk? Zo beschrijven ze zichzelf:

“CoderDojo is een open source, door vrijwilligers opgezette beweging die staat voor het houden van gratis, non-profit development clubs en reguliere sessies voor jonge mensen. Omdat CoderDojo open source is kan elke Dojo anders zijn en staat volledig los van andere Dojo's.

Bij een Dojo leren jonge mensen, tussen de 5 en 17 jaar, om te programmeren, websites bouwen, apps ontwikkelen, programma's te maken, games te maken en meer. Dojo's worden opgezet en georganiseerd door vrijwilligers. Naast het leren programmeren, ontmoeten deelnemers ook anderen like-minded mensen, delen deelnemers waar ze aan aan het werken zijn en maken ze nieuwe dingen. CoderDojo maakt development en leren programmeren een leuke, sociale en kick ass ervaring. CoderDojo legt ook veel nadruk op open source en gratis software en heeft wereldwijd een breed netwerk aan vrijwilligers. CoderDojo heeft slecht één regel: "Boven Alles: Wees Cool!", pesten, liegen en mensen hun tijd verdoen is niet cool. Om aan een Dojo mee te doen kun je er een bij jou in de buurt zoeken of zelf een opstarten.”

CoderDojo Maaspoort heeft een vaste stek bij 'De Plek' in de sporthal bij FlikFlak. Momenteel liggen de activiteiten stil vanwege corona, maar als het je leuk lijkt kun je hun site in de gaten houden om te zien wanneer ze weer van start gaan.

<https://www.coderdojo-denbosch.nl/maaspoort/>