



Twee drukknop bediende schakelaars



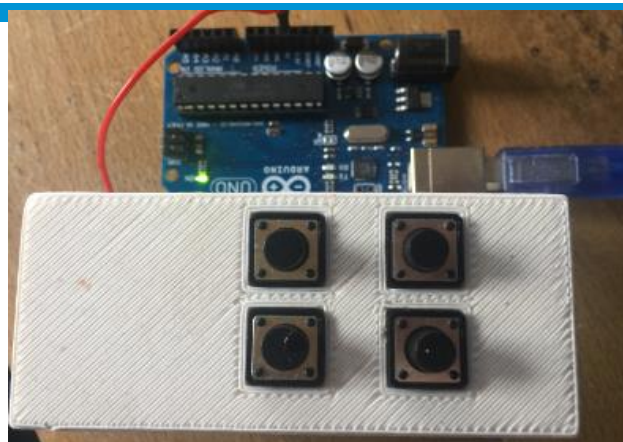
Voorkennis:

Les 1 t/m 10 en Dobbelsteen

Leerdoelen: Maken van parallele taken

Benodigdheden

- Arduino (UNO)
- LCD4DLS unit



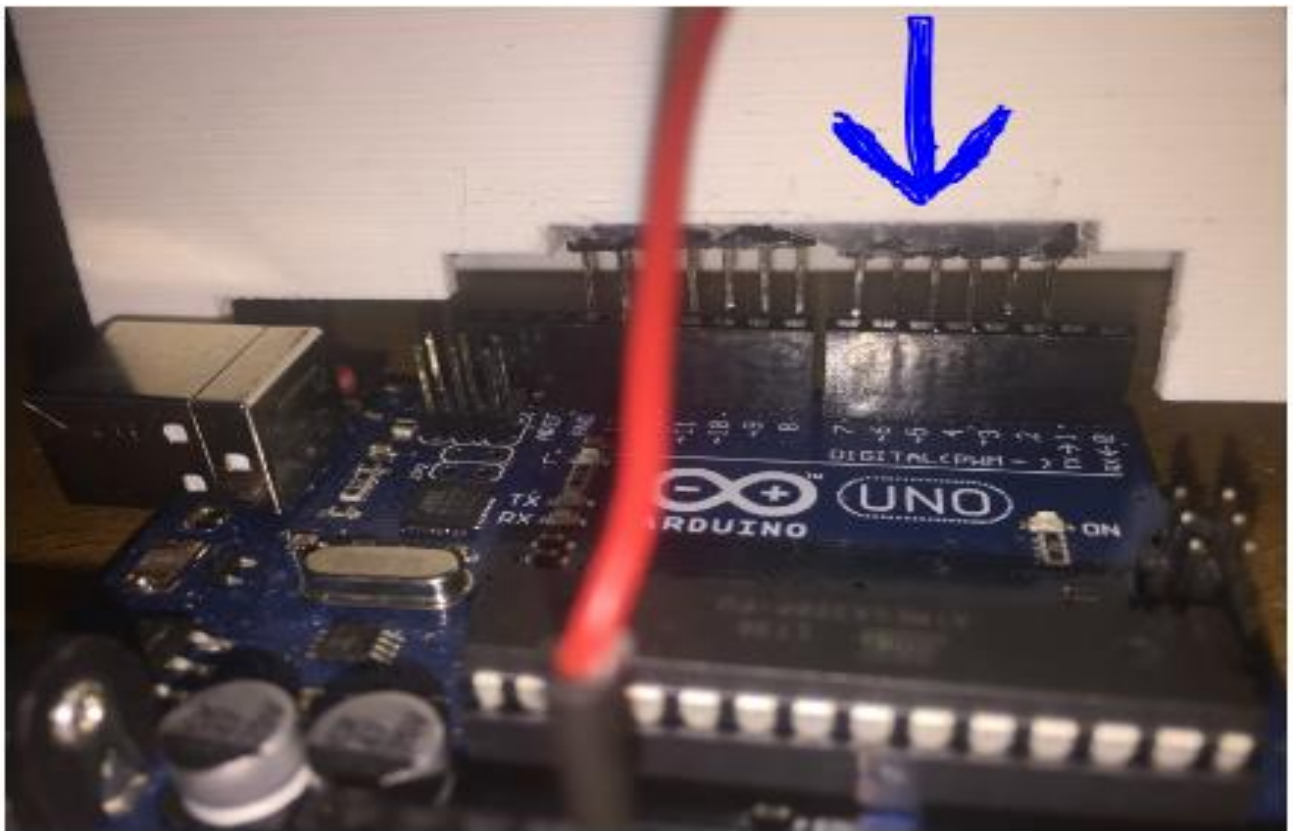
Introductie

Je gaat ter voorbereiding op de opdrachten **Stopwatch**, **Eierwekker** en **Stroboscoop** twee drukknop bediende schakelaars maken met één arduino. Je maakt gebruik van de LCD4DLS unit. De **LCD4DLS** unit heeft een **LCD**, **4 Drukschakelaars**, een **LED** en een **Speakertje** en is eenvoudig en snel aan te sluiten op de arduino uno. Voor deze opdracht maak je alleen gebruik van de drukknoppen, de LED en het speakertje.

Bij de **LCD4DLS** unit behoort een sketch (software) waarin is vastgelegd met welke IO poorten van de Arduino het LCDisplay, de **4 Drukknoppen** en de **Led** zijn verbonden en is een deel van de nodige initialisatie reeds verzorgd. Bijvoorbeeld het instellen van de Digital IO poorten voor de drukknoppen als **INPUT**. Kopieer de inhoud van "**LCD4DLS.txt**" naar de arduino IDE. Dit bestand kan je vinden op de website van www.coderdojo-zoetermeer.nl onder opdrachten en → Arduino → AdvancedProjects.



De LCD4DLS unit aansluiten



De LCD4DLS unit heeft aan de achterkant een Pin Header waarmee je bijna alle verbindingen in één keer kan maken. Zorg er voor dat alle pinnetjes van de Pin header goed tegen over de gaatjes van de arduino zitten, 6 in de ene connector en 5 in de andere connector (zie de **blauwe pijl** in de foto hierboven).

Daarnaast moet één **RODE draad** op de "5V" volt worden aangesloten.

Het uitlezen van drukknoppen.

Eerst een kleine uitleg hoe de arduino de software doorloopt. Zoals in les #1 Blink is aangegeven zijn er minimaal twee functies, te weten `setup()` en `loop()`. De functie `setup()` wordt éénmaal uitgevoerd als de arduino opstart/reset. De functie `loop()` wordt steeds opnieuw uitgevoerd als de laatste instructie in die functie is uitgevoerd.

In les #3 KnopAanUit heb je geprogrammeerd dat als de knop ingedrukt is, de LED ook aan is. In de functie `loop()` wordt steeds opnieuw de drukknop "uitgelezen" met de code "if (`digitalRead`(KnopPin) == HIGH)". Dit uitlezen gebeurt heel veel malen per seconden. Voor die opdracht was dat uitstekend, maar in deze opdracht mag het éénmaal indrukken van één van de bovenste drukknoppen maar één status verandering geven.

Je moet dus iets verzinnen om te bepalen of de knop nog steeds of opnieuw is ingedrukt als de drukknop is uitgelezen.



OPDRACHT

Maak twee drukknop bediende schakelaars

De twee drukknop bediende schakelaars moet kunnen:

- Als er op de rechts_boven knop wordt gedrukt en de LED **uit** is, gaat de LED **aan** totdat er weer **opnieuw** op de rechts_boven knop wordt gedrukt.
- Als er op de links_boven knop wordt gedrukt en de speaker **uit** is en gaat de speaker **aan** totdat er weer **opnieuw** op de links_boven knop wordt gedrukt.

Het doel van deze opdracht is leren maken van (virtueel) parallelle taken, dwz dat de arduino niet actief wacht op een gebeurtenis maar door gaat met andere taken en "later nogmaals kijkt" of de gebeurtenis heeft plaats gevonden. De CPU van een arduino is snel en kan vele instructies (ca 1.000.000) in één seconde uitvoeren. Dus dat "later nogmaals kijkt" kan al binnen 1 milli seconden zijn. Dit hangt af natuurlijk af van het aantal instructies in de andere taken.

Bij de **LCD4DLS** unit behoort een stukje software waarin is vastgelegd met welke IO poorten van de Arduino het LCDisplay, de **4 Drukknoppen** en de **Led** zijn verbonden en is een deel van de nodige initialisatie reeds verzorgd. Bijvoorbeeld het instellen van de Digital IO poorten voor de drukknoppen als **INPUT**.

Kopieer de inhoud van de file **LCD4DLS.txt** en plaatst dat in de IDE van Arduino en sla (save) deze sketch onder de naam duoknoppen. Indien nodig, vraag hulp van de mentor.

In de sketch (het programma dat je net hebt gekopieerd) hebben de drukknoppen de namen: "rechts_boven", "rechts_onder", "links_boven" en "links_beneden".

De input van een knop is "**HIGH**" als de knop is ingedrukt en "**LOW**" als de knop niet wordt ingedrukt.

Maak twee taken.

1. Bepaal en administreer dat de drukknop links_boven van niet ingedrukt overgaat naar wel ingedrukt en zet de speaker aan of uit
2. Bepaal en administreer dat de drukknop rechts_boven van niet ingedrukt overgaat naar wel ingedrukt en zet de LED aan of uit.

Zie de hints op de volgende pagina!



HINTS

Declareer voor iedere bovenste drukknop een variabele van het type **boolean** en initialiseer die variabelen met de waarde **“false”** .

Geef die variabelen de waarde **“true”** als je detecteert dat de overeenkomstige drukknop van niet ingedrukt over gaat naar ingedrukt en **“false”** als je detecteert dat de overeenkomstige drukknop niet meer is ingedrukt.

Om de juiste status verandering van de LED te kunnen uitvoeren, zal je dus ook de status (**aan** of **uit**) van de LED en van de speaker moeten administreren. Ook dit kan met variabelen van het type boolean.

Geef de variabelen duidelijke namen, dus niet var1, var2, var3 enz, maar:

- rechts_boven_ingedrukt
- links_boven_ingedrukt
- led_aan
- speaker_aan

Het lijkt nu meer werk om al die lange namen voor de variabelen in te tikken, maar als volgende week (of de volgende keer) verder gaat met deze sketch weet je niet meer wat er in de variabele var2 wordt geadministreerd maar dat weet je nog wel van de variabele links_boven_ingedrukt.

Met de functies **“analogWrite(speaker,127)”** en **“digitalWrite(speaker,LOW)”** kan je de speaker **aan** of **uit** zetten.



Uitbreiding 1

De functie `setup()` wordt eenmaal uitgevoerd maar hoe vaak wordt de functie `loop()` uitgevoerd?

Maak nog een taak die weergeeft op de seriële monitor het totaal aantal keren dat de functie `loop()` is doorlopen en hoe vaak de functie `loop()` is doorlopen in de afgelopen seconde.

Declareer twee variabele `unsigned long loop_teller` en `unsigned long loop_teller_oud`. Verhoog de variabele "loop_teller" met 1 aan het begin van de functie `loop()`.

Maak nog een taak die na iedere seconde de waarde van de variabele "loop_teller" naar de serial monitor print met de instructies "`Serial.print("loop_teller = ")` en `Serial.println(loop_teller)`". Bereken het aantal keren dat de functie `loop()` in de afgelopen seconde is doorlopen en print dat ook naar de seriële monitor.

De arduino houdt bij hoeveel milliseconden zijn verstreken sinds de arduino is gereset/aangezet. Het aantal milliseconden kan worden opgevraagd met de functie `millis()` en deze waarde is van het type `unsigned long`. Een variabele `unsigned long` kan een waarde hebben van 0 tot en met 4.294.967.295.

Vraag 1

Je zou nu moeten zien dat de functie `loop()` ongeveer 190.000 keren per seconde wordt doorlopen.

Klopt dat? Zo niet, vraag hulp!

Vraag 2

Heb je gemerkt als je maar éénmaal de knop hebt ingedrukt het lijkt alsof je de knop tweemaal hebt ingedrukt (led blijft aan of blijft uit)?

In de drukknop zit een vast en een beweegbaar contact. Als je de knop indrukt gaat het beweegbare contact naar het vaste contact en kan er een stroom gaan lopen. Maar soms stuitert het beweegbare contact heel even terug. Dus kan er heel even een stroom lopen en dan heel even niet en daarna weer wel. Dit gaat heel snel en dit verschijnsel heet denderen.

Je hebt gezien dat de ongeveer 190.000 keer per seconde de functie `loop()` wordt doorlopen. Dus ook de drukknoppen worden 190.000 keer per seconde uitgelezen. Dat is $1.000.000 / 190.000$ is ongeveer iedere 6 micro seconden. Is het nodig om dit zo vaak te doen? Nee dus. Zorg er voor de taken die de knoppen uitlezen maar 100 keer per seconden worden uitgevoerd.

Hiermee wordt het denderen probleem opgelost.



Uitbreiding 2

LCD aansturen

Zie <https://www.arduino.cc/en/Reference/LiquidCrystal> voor de aansturing van het LCD.

Voeg instructies toe aan de taken die de knoppen uitlezen zodat ook de status van de led en speaker worden weergegeven op het LCD display.





Nieuwe uitdaging

Nieuwe uitdaging

Ga verder met de sketch “duodrukknoppen2”.

De nieuwe opdracht is dat twee drukknoppen tegelijkertijd ingedrukt moeten zijn om een status verandering (LED **uit** → LED **aan** of LED **aan** → LED **uit**) door te voeren. Een nieuwe status verandering mag pas plaats vinden nadat beide drukknoppen tegelijkertijd niet ingedrukt zijn geweest.

Maak een nieuwe taak die beide onderste knoppen “tegelijkertijd” uitleest.

LET OP

Deze sketch is het startpunt voor onder andere de opdrachten Stopwatch en Stroboscoop, dus deze sketch goed bewaren!

Maak met de arduino IDE een kopie van je deze sketch met “Save as... Ctrl+Shift+S” en noem die “stopwatch”.