

Voorkennis	Les 1 t/m 10	Niveau	
Benodigheden	Arduino en LCD4DLS unit	Leerdoelen	Maken van parallele taken

Twee drukknop bediende schakelaars



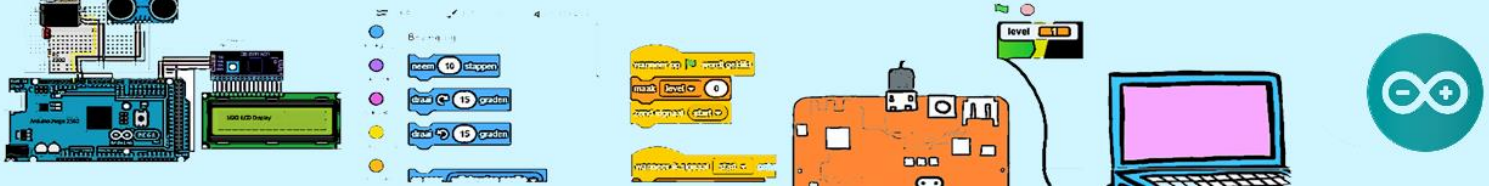
1. Introductie

Je gaat ter voorbereiding op de opdrachten **Stopwatch** 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.ino**" en plaats dat in de IDE van Arduino en sla (save) deze sketch onder de naam duoknoppen. In de sketch (het programma dat je hebt opgeslagen als duoknoppen) hebben de drukknoppen de namen: "rechts_boven", "rechts_onder", "links_boven" en "links_beneden". 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 geadmistriseerd maar dat weet je nog wel van de variabele links_boven_ingedrukt. Indien nodig, vraag hulp van de mentor.

Dit bestand kan je vinden op de website van coderclub.nl onder Arduino.

Het doel van deze opdracht is om één arduino meerdere taken, schijnbaar onafhankelijk van elkaar te laten uitvoeren. Een arduino kan maar één instructie tegelijk uitvoeren, maar een arduino is heel snel. Hij kan z'n 1 miljoen



instructies per seconde uitvoeren. Omdat de arduino zo snel is, lijkt het voor een mens dat de instructies (taken) parallel worden uitgevoerd. Maar dan mogen we geen taken maken die de processor lang achter elkaar bezet houden. Je kan dus geen gebruikmaken van de delay() functie. Als je met de do ...while (digitalRead(modeknop)) wacht totdat een knop niet meer wordt ingedrukt en de persoon drukt 2 seconden op de knop, dan kan de processor 2 seconden niets anders doen. Hieronder in “3. Het Uitlezen van drukknoppen” wordt uitgelegd hoe dat kan zonder de processor bezet te houden.

2. 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 tegenover 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.

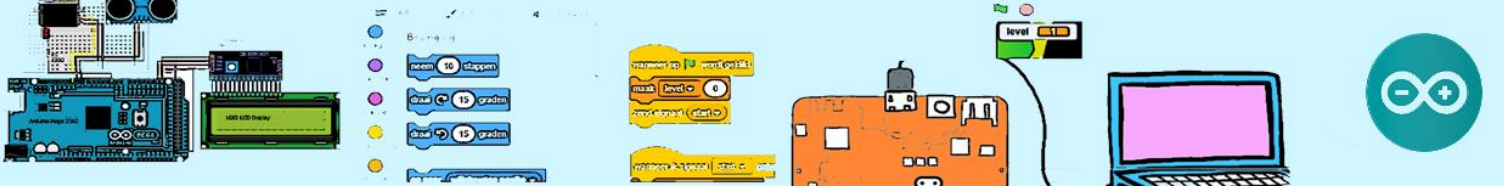
3. Het uitlezen van drukknoppen.

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))”. Dit uitlezen gebeurt heel veel malen per seconden. Voor die opdracht was dat uitstekend. In deze opdracht moet het éénmaal indrukken van de linker_boven drukknop de speaker gaan piepen bij nogmaals indrukken van de linker_boven drukknop de speaker niet meer piepen.

We moeten dus iets verzinnen om te bepalen of de knop nog steeds of opnieuw is ingedrukt als de drukknop wordt uitgelezen.

Allereerst gaan we gaan een boolean (knop_ingedrukt) declareren waarmee we gaan bijhouden of de drukknop al was ingedrukt voordat we controleren wat de huidige status is. Klinkt moeilijk maar dat is het niet. Zie hier onder.

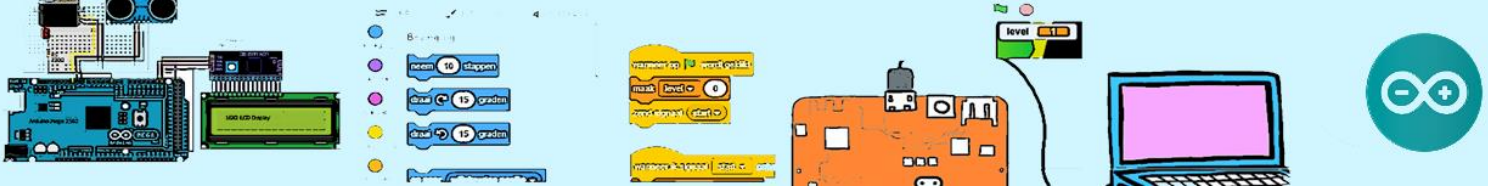


```
if (!links_boven_ingedrukt) // links_boven knop was laatste keer niet ingedrukt
{
  if (digitalRead(links_boven)) // knop nu ingedrukt?
  {
    links_boven_ingedrukt = true;
    doe wat nodig is // hier programmeren wat er moet gebeuren als de knop is ingedrukt
  }
}
else //(links_boven_ingedrukt) // knop was dus al ingedrukt
{
  if (!digitalRead(links_boven)) // is de knop niet meer ingedrukt?
  {
    links_boven_ingedrukt = false;
  }
}
} //endif(!links_boven_ingedrukt)
```

Een tweede uitdaging is het volgende, namelijk denderen van het drukknopcontact. Als vergelijking: als je een bal op de grond laat vallen, dan zal de bal stuiten. Dit zelfde geldt voor het drukknopcontact. Als je op de drukknop drukt, gaat het drukknopcontact tegen het vaste contact, maar zal heel even stuiten en dan weer contact maken. En dat zal zich een paar maal herhalen. En dit alles gebeurt binnen 10 mille seconden. Als een mens op een knop drukt, dan is een responsetijd van 0,1 seconde meer dan voldoende, terwijl de CPU ca 100.000 instructies in die 0,1 seconde uitvoert. Een manier om het denderen probleem op te lossen is om maar eenmaal in de 0,1 seconden de status van de drukknop uit te lezen. 0,1 seconde is 100 mille seconden, dus de knoppen_lees_periode = 100. En als we dat combineren met bovenstaande dan krijgen we:

```
if (huidige_milli_seconden > knoppen_lees_tijd) // proces die de knoppen leest
{
  knoppen_lees_tijd += knoppen_lees_periode;
  if (!links_boven_ingedrukt) // links_boven knop was laatste keer niet ingedrukt
  {
    if (digitalRead(links_boven)) // knop nu ingedrukt?
    {
      links_boven_ingedrukt = true;
      doe wat nodig is // hier programmeren wat er moet gebeuren als de knop is ingedrukt
    }
  }
  else //if(links_boven_ingedrukt)
  {
    if (!digitalRead(links_boven)) // knop niet meer ingedrukt
    {
      links_boven_ingedrukt = false;
    }
  }
} //endif(!links_boven_ingedrukt)

} // endif huidige_milli_seconden > knoppen_lees_tijd
```



4. Opdracht: Maak twee drukknop bediende schakelaars

De twee drukknop bediende schakelaars moeten kunnen:

- Als er op de links_boven knop wordt gedrukt en de speaker **uit** is en gaat de speaker **aan**.
- Als er op de links_boven knop wordt gedrukt en de speaker **aan** is en gaat de speaker **uit**.
- Als er op de rechts_boven knop wordt gedrukt en de LED **uit** is, gaat de LED **knipperen** met 1 seconden aan en 1 seconden uit en dan weer 1 seconden aan en 1 seconden uit enz.
- Als er op de rechts_boven knop wordt gedrukt en de LED **knippert** is, gaat de LED **uit**.

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**.

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

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

Maar hoe laat je nu de LED knipperen? HINT: maak een nieuwe taak.

5. Uitbreiding

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.

