



Arduino Stopwatch



Voorkennis:
Duoknoppen

Leerdoelen: parallel processing

Benodigheden

- Arduino (UNO)
- LCD4DLS unit
- Kopie van de sketch duoknoppen



Introductie

Je gaat een **stopwatch** maken. Je maakt gebruik van de LCD4DL 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 de stopwatch maak je alleen gebruik van het LCD en de drupknoppen. Bij de opdracht Stopwatch maak je gebruik van de het programma (sketch) dat je gemaakt hebt gemaakt bij de opdracht Duoknoppen. Als je nog geen kopie hebt dan laad deze sketch in de Arduino IDE en sla het op als "Stopwatch".

In de opdracht "Duoknoppen" heb je geleerd om bijvoorbeeld de LED aan of uit te zetten als op een knop wordt gedrukt. Nu moet je niet de LED aan of uit zetten maar de stopwatch starten of stoppen.. Als op de andere knop wordt gedrukt, wordt de ronde tijd vastgelegd of om de stopwatch te resetten. Je gaat het LCD gebruiken op de verstreken tijd en ronde tijd weer te geven.

De stopwatch moet kunnen starten, stoppen, doorstarten, resetten en de ronde tijd ondersteunen (zoals vergelijkbaar op je telefoon).



OPDRACHT

Maak een stopwatch

De stopwatch moet kunnen:

- Starten,
- Stoppen,
- Doorstarten,
- Resetten,
- Ronde tijd ondersteunen (zoals vergelijkbaar op je telefoon).

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

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, en dat is $(4.294.967.295 / (3.600.000\text{sec} * 24\text{uur}))$ bijna 50 dagen.

Voor deze stopwatch is dat lang genoeg.

Wat jij dus moet doen is:

1. Je moet dus iets verzinnen om te bepalen of de versterken milliseconden wel of niet wel of niet tot de gemeten tijd behoren.
2. De verstreken tijd netjes naar het LCD printen. Dus in uren:minuten:seconden,milliseconden bijvoorbeeld 00:01:12,231.

Hints:

- Declareer een variable "stopwatch_meet" van het type **boolean** die aangeeft of de stopwatch de tijd wel of niet meet en declareer een variabele "stopwatch_tijd" van het type **unsigned long** waarin de verstreken milliseconden worden opgeslagen.
- Zowel de stopwatch tijd en ronde tijd moet worden weergegeven op het LCD. Maak één functie die de tijd weergeeft maar de meegegeven parameters (rij en kolom) bepalen de plaats op het display. De stopwatch tijd op de onderste rij zoveel mogelijk naar rechts en de rondetijd op de bovenste rij zoveel mogelijk naar links.
- Het heeft alleen zin om de stopwatch tijd opnieuw weer te geven als deze verandert is sinds de vorige weergave. Dus `if (stopwatch_tijd != stopwatch_tijd_oud)` pas opnieuw weergeven.
- Zoals eerder aangegeven, gebruik betekenis volle namen voor variabelen en functies.



Verbetering

Wordt de loop teller nog steeds weergegeven? Zo niet, maak dat weer werkend.

Meet de tijd in microseconden die nodig is om de stopwatch tijd op het display te zetten en geef deze tijd weer op de serial monitor.

Vraag: Hoelang duurt het?

Als het goed is duurt dit ongeveer tussen de 4300 en 4700 microseconden. En dat is dus ruim 4 milliseconden.

Als de Arduino processor ruim 4 milliseconden nodig heeft om de tijd weer te geven, kunnen we dus niet een stopwatch maken die één milliseconde nauwkeurig is. Dus laten de stopwatch aanpassen zodat deze op een honderdste seconde nauwkeurig is.