

Algoritmer och datastrukturer

Projekt Squashslag

Projektrapport

Grupp 33

Dylan Saleh

Johan Kämpe

Projektbeskrivning

Projektets syfte är att registrera olika slagtyper som en squashspelare kan göra med ett racket, och logga dessa som en tabell i en textfil. Tabellen ska sorteras efter olika värden:

- Tidpunkt för slaget
- Antal slagtyper
- Slagkraft/hastighet

Konstruktion

Registrering av slag kommer utföras genom att utrusta ett squash-racket med en gyro- och accelerometer-sensor. Data från sensorn behandlas av en Arduino-enhet, och textfilen lagras på ett SD-kort som sitter i en SD-kortsmodul.

Testupplägg

Initialt i projektet kommer tester att utföras genom att göra flertalet slag i luften med Arduino-enheten kopplad till en dator. Data från accelerometern studeras via *Serial monitor* i Arduino-IDEn.

Mjukvarukrav

Mjukvaran ska kunna lagra slag i en struct array, som innehåller tid, slagtyp, slagtidslängd, och slagstyrka. Olika sorteringsalgoritmer kan komma att användas, bedömning av dessas prestanda kan göras genom att skriva ut tiden för sortering i textfilen.

Avgränsningar

Endast slagtyperna forehand och backhand kommer att loggas.

För att enklare kunna avgöra när ett slag har utförts antas att racket är stilla under en kort period. Dvs. att man har en pausperiod mellan slagen.

Genomförande

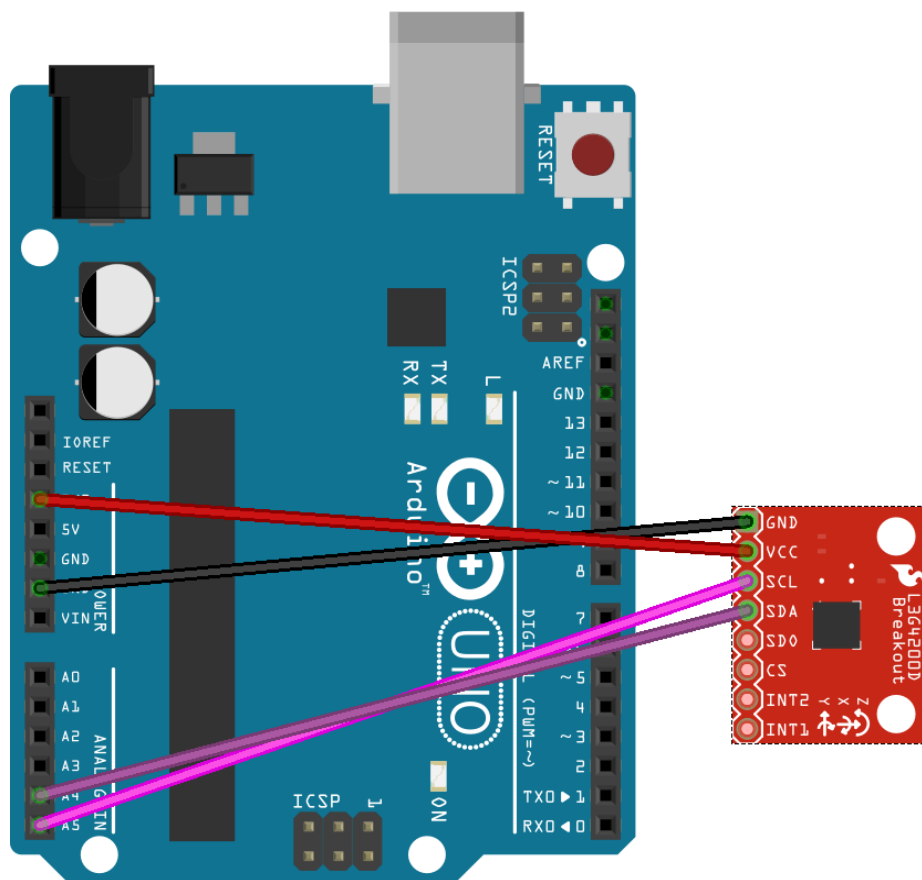
Uppkoppling och komponenter

Den slutgiltiga konstruktionen är sammanställd av en arduino och en MPU-enhet som innehåller en gyro- och en accelerometer-sensor.

Komponenter:

- Utvecklingskort: Arduino Uno
- Gyro/Accelerometer: MPU-9250
- Minneskortläsare: Micro-SD Breakout board+

Fritzing-schema:



fritzing

Datastrukturen och insamling av data

Data samlas in till en struct array, med storleken 100. Structens medlemmar är:

- `int16_t styrka;`
- `int tid;`
- `int langd;`
- `bool typ;`
- `int slagnummer`

Variabeln **slagnummer** sätts till det aktuella utförda slagets nummer i slagserien, det första slaget som loggas sätts till 1, det andra till 2, osv.

Variabeln **typ** sätts till 1 eller 0, beroende på om det är ett backhand- eller forhand-slag som loggas. Slagets typ avgörs genom att läsa av accelerometers värde för acceleration i dess Z-riktning. Om accelerations-kurvan startar negativt anses slaget vara av typen *forehand*, annars *backhand*.

Variabeln **tid** håller data för när ett slag har utförts, mätt i millisekunder efter att Arduino-enheten har startat dess program. Variabeln **langd** (längd) håller data för hur långt ett slag är tidsmässigt, i millisekunder.

Värden för **tid** och **langd** avgörs med Arduino-standardfunktionen **millis()**, som returnerar ett värde i millisekunder, efter att det aktuella inladdade programmet startar i Arduino-enheten.

Variabeln **styrka** håller det värde som anses vara slagets slagkraft, detta tas från absolutvärdet av accelerometers värde för acceleration i dess Z-riktning. Variabeln sätts när ett slag triggar.

Modifikation av kod

Projektets kod har skrivits så att slag kan registreras mycket fort från accelerometer-värdena. Detta för att kunna fylla struct-arrayen fort inför varje nytt test. I stället för att slå med racketet, så vickades på accelerometer-enheten med manuellt med handen.

Kalibrering av accelerationsvärde

Initialt visas inte data från MPU-enhetens accelerometer korrekt. Värdet som erhålls har oftast ett högre värde än noll, även när enheten är helt stilla. Detta korrigeras genom användning av en egenbyggd kalibrering, vid programmets uppstart.

Granskning av utdata

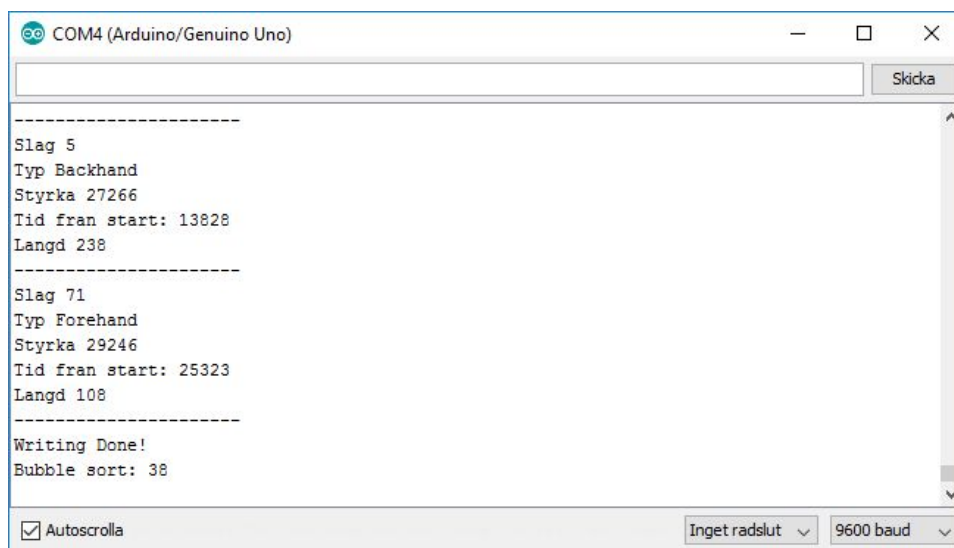
Ursprungligen var tanken att data skulle skrivas till en textfil på ett SD-kort. Dock kunde inte SD-kortläsaren användas tillsammans med accelerometern, på grund av tidsbrist valdes istället att studera utdatan i Arduino-IDEns serial monitor, där textutskrift från Arduino-enheten kan visas.

I serial monitor skrivs värden för varje loggat slag ut. När max antal slag har registrerats sorteras arrayen efter slagstyrka, arrayen skrivs sedan ut i den nya sorterade ordningen, med tiden för sorteringen sist.

Studie och val av sorteringsalgoritm

I projektet används och jämförs sorteringsalgoritmerna *bubble sort*, *insertion sort* och *selection sort*.

Inför varje nytt sorteringstest fylls en ny array med nya slag. Tider för sorteringsalgoritmer:

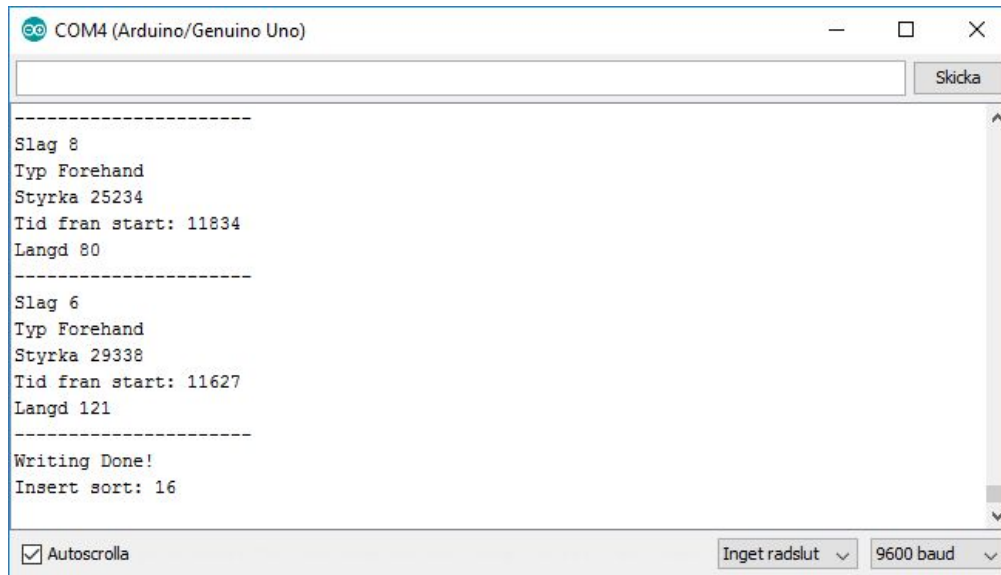


The screenshot shows the Arduino Serial Monitor window titled "COM4 (Arduino/Genuino Uno)". It displays two sets of data for tennis strokes, separated by dashed lines. The first set is for "Slag 5" (Typ Backhand) with a strength of 27266, start time of 13828, and length of 238. The second set is for "Slag 71" (Typ Forehand) with a strength of 29246, start time of 25323, and length of 108. Below the data, it says "Writing Done!" and "Bubble sort: 38". At the bottom, there are checkboxes for "Autoscrolla" (checked), a dropdown for "Inget radslut", and a dropdown for "9600 baud".

```
COM4 (Arduino/Genuino Uno)

-----
Slag 5
Typ Backhand
Styrka 27266
Tid fran start: 13828
Langd 238
-----
Slag 71
Typ Forehand
Styrka 29246
Tid fran start: 25323
Langd 108
-----
Writing Done!
Bubble sort: 38

[Autoscrolla] [Inget radslut] [9600 baud]
```



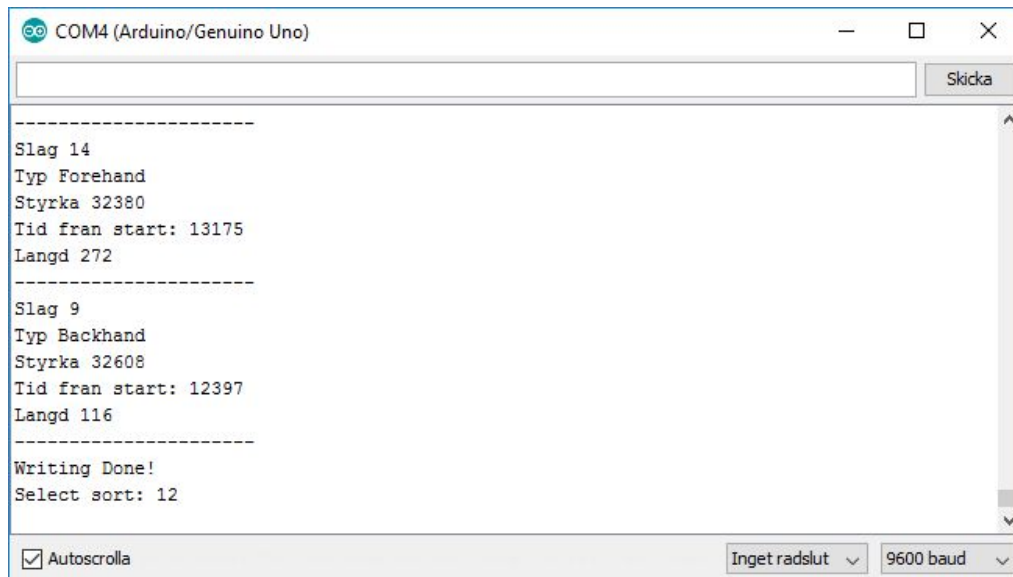
COM4 (Arduino/Genuino Uno)

Slag 8
Typ Forehand
Styrka 25234
Tid fran start: 11834
Langd 80

Slag 6
Typ Forehand
Styrka 29338
Tid fran start: 11627
Langd 121

Writing Done!
Insert sort: 16

☒ Autoscrolla Inget radslut 9600 baud



COM4 (Arduino/Genuino Uno)

Slag 14
Typ Forehand
Styrka 32380
Tid fran start: 13175
Langd 272

Slag 9
Typ Backhand
Styrka 32608
Tid fran start: 12397
Langd 116

Writing Done!
Select sort: 12

☒ Autoscrolla Inget radslut 9600 baud

Algoritmen *quicksort* valdes att inte testas, då denna anses inte vara lämplig för Arduino och små arraystorlekar.

Det anses inte spela roll vilken av algoritmerna som används, sorteringstiden för detta projekt är inte av hög vikt. Samtliga sorteringstider är snarlika.

Algoritmerna använder sig av samma extra variabler; två int-variabler som håller index för arrayerna, samt en temp struct för byte mellan två arrayplatser.

Ändringar från projektplan

- Inget SD-kort eller textfil användes för att skriva ut data.
- Endast slagstyrkan används för sortering
- Insamling av slag utförs inte genom att slå med ett racket
- Slagnummer tillades till structen för att bättre kunna se att arrayen har sorterats korrekt.

Slutsats & diskussion

Projektet krävde inget behov av snabbhet för sorteringsalgoritmerna, och därmed testades tre olika sorteringsalgoritmer (selection sort, insert sort och bubble sort) som är relativt enkla i sin komplexitet och passande för mindre arrayer.

Selection sort valdes ändå som sorteringsalgoritm för projektet, baserat på att den var snabbast. Även om detta inte var en viktig faktor.