Projekt B

Hardball projekt:
Tilføjelse til spillet og kampens hede i form af Tripwire alarm

Afleveringsfrist: 24/05-2023

Tegnoptælling: 34.897

Elev: Nicolai Schou –
52765@student.eadania.dk

Opsummering:

I samarbejde med virksomheden Aarhus Eventpark er vi kommet frem til at der kunne være flere ting der kunne tilføjes til deres spil Hardball, de har en indendørs bane der gør brug af hjørner, snævre gange og mindre åbne rum spredt ud på banen, til mødes med virksomheden blev der brainstorme en del i forhold til hvad vi kunne tilføje til banen der kunne rykke på spillets rammer. Der blev talt om at designe og lave en turret med motion sensor, men også en skydeplade til at sende røg ud i en gang eller et rum. Det som vi kom frem til, var en Tripwire alarm. En tripwire alarm kan laves på en Arduino Uno og kan laves meget kompakt hvilket kun er en fordel. Ideen med en tripwire alarm er at placere en eller flere rundt omkring på banen, de skal være fire and forget. Så Arduino Uno skal fodret med koden den skal køre nonstop, den skal have sine komponenter tæt pakket og eller skal den bare sættes op, helst hvor man ikke bemærker den, der var også tale om at den skal være transportabel i forhold til at hvert hold i spillet kan gøre brug af dem i forhold til enten at dække deres egen ryg eller opspore fjendens positioner.

Indholdsfortegnelse:

Liste over figurer:

Definitioner / Forkortelser:

IOT	Internet Of Things
C++	Programmeringssprog
Arduino	Mikrocontroller
Tripwire	En snor designet til at fælde en fjende
Alarm	Når der sker noget der ikke skal ske, giver den en høj lyd
OS	Operativ System
GND	Ground - jordforbindelse

Introduktion:

Hardball eller Airsoft er en sportsgren der sætter to eller flere hold til at kæmpe mod hinanden i forskellige spil og på forskellige baner. Der bliver brugt hardball våben der afhængig af udgangskraft (gas, el), bliver målt i meter per sekund eller joule som man må skyde inden for en given sikkerhedsafstand. Spillet vindes afhængigt af reglerne for det givne spil, normalt ved eliminering eller erobring af modstanderens flag.

Hardball stammer fra Japan, Ichiro Nagata udviklede et våben, hvis funktion var hverken at dræbe eller fungere som en replika. På denne måde kunne det overholde de meget strenge våbenregler i Japan i den tid. Det første hardball våben så dagens lys i de tidlige 1970'ere og blev markedsført som en "soft air gun". Navnet "soft air gun" stammer fra våbnets første drivmiddel svarende til en komprimeret Freon-silicone oile. Senere blev den erstattet af en propanbaseret silikone oile eller "grøn gas". Oprindeligt blev våbnet brugt som et målskydningsvåben. Det var først senere man begyndte at skyde efter andre spillere ligesom i paintball. I 80'erne og tidligt 90'ere kom hardball til Europa, det blev introduceret i Storbritannien. Her blev de solgt i dele, hvormed det var nødvendigt at samle dem. Siden midten af 80'erne har man spillet hardball som en rekreationel hobby for alle aldre. Det første elektriske hardball våben eller AEG (automatic electric gun) blev udviklet af japanske Tokyo Marui i 92. Hovedparken af alle våben produceres i Asien.[1]

Hardball har eksisteret i mange år og har rykket sig adskillige gange i forhold til hardware, i dag har man alt fra granater, til flere forskellige typer af våben, alt sammen baseret på virkelighedens våben. Dog har man ikke set meget nyt i forhold til sporten og hvordan spillets gang går i lang tid. Dette projekt vil kigge på hvordan man kunne tilføje til sporten i form af ekstra udstyr eller hardware der kan give et hold en edge eller fordel i forhold til deres modstander. Princippet kunne være først til mølle eller måske som en krykke hvis slaget går i den forkerte retning. Dette vil jeg komme meget mere ind på i løbet af rapporten, hvor vi kommer til at forstå mere om hvad man kunne implementere og hvordan det kan ændre spillets tilgang.

[1] https://da.wikipedia.org/wiki/Hardball

Problemstilling:

Hvordan kan en tripwire alarm øge mulighederne i Hardball spil.

- Hvorfor kunne ny hardware som en tripwire alarm flytte på spillets rammer?
- Er det muligt at udvikle en tripwire alarm der kan holde til spillets omstændigheder?
- Hvilke begrænsninger er der i forhold til en tripwire alarm og dens implementering i spillet?

Begrænsninger:

Hardball er en sportsgren der både foregår inde såvel som ude afhængig af sæsonen, dog er der store klubber der gerne bruger skove, andet udendørs terræn året rundt. Dette kan tilføje til udfordringerne i at skulle udvikle ny hardware eller udstyr til spillerne, da man skal have vejrforhold og tilsvarende med i sine tanker når man skal designe og udvikle et produkt. Grundet tidsudfordringer kan projektet kun udføres teoretisk med en mulig dog ikke specielt funktionel prototype.

Da Aarhus Eventpark udnytter en indendørs bane undgår vi at skulle tænke over hvordan man sikrer vores komponenter mod udendørs brug, der vil der stadig være en eftertanke omkring udendørs brug da dette kunne blive en mulighed.

Der vil kun være teoretisk viden og en hardware opsætning med software som påviser hvordan man kunne udvikle en tripwire alarm med en Arduino Uno og ideen om fire and forget. Det ville kræve materialer som der ikke er tid eller penge til at producere hvis der skulle laves en case eller en form for beskyttelse til komponenterne som vi skal gøre brug af.

For at skulle producere de materialer som skulle bruges, ville der enten skulle være mulighed for brug af træ, lette metaller eller en 3d printer. Afhængig af holdbarheden ville man skulle bruge en af disse materialer. Hvis det skulle gøres nemt og simpel, og ligetil at reparerer hvis der sker skader på casen, så ville en 3d printer være vejen frem, da man nemt ville kunne lave de forskellige komponenter til en omsvøbende case som kan beskytte med mindre slag, tab eller udendørs elementer. Man skal også have muligheden for at tripwire alarmen kan blive ramt af et skud fra et hardball våben under spillet. Til dette ville tungere og mere stabile materialer være nødvendige da dele fra en 3d printer kun ville give minimal beskyttelse i forhold til dette. Træ eller lette metaller ville være vejen frem, men så skal man enten selv behandle det eller få en anden virksomhed til at udarbejde et design og producere en case. Dette ville koste penge og der er ikke givet nogle penge til projektet.

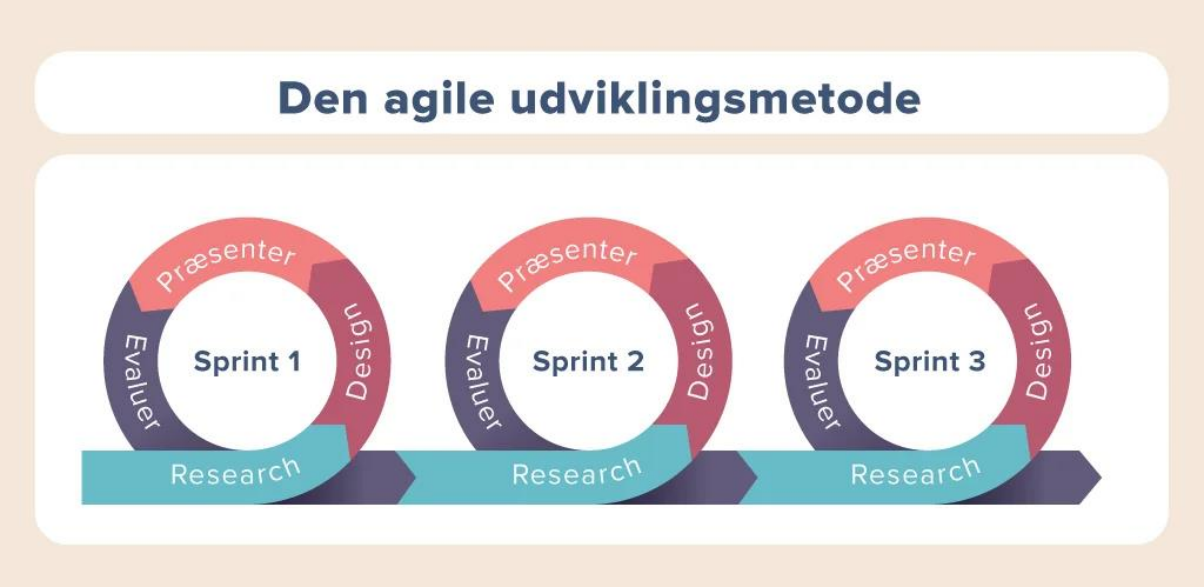
Alt i alt så er der en del begrænsninger i forhold til dette projekt, havde man haft mere tid og penge til at udvikle mere så ville en funktionel tripwire alarm ikke være umuligt at producere og gøre brug af indenfor spillets rammer.

Metoder og Værktøjer:

Arbejdsmetoder:

Agile-metoden:

I dette projekt har jeg valgt at gøre brug af Agile-metoden, den er kendt for sin fleksibilitet og gør det nemt at tilpasse sig ændringer undervejs i projektet. Ved at inddrage Aarhus Eventpark i hver milepæl, kan man få feedback og eventuelle ændringsønsker undervejs, på den måde kan man sikre at det endelige resultat imødekommer deres krav og forventninger.



Figur 1 - Agile-metoden

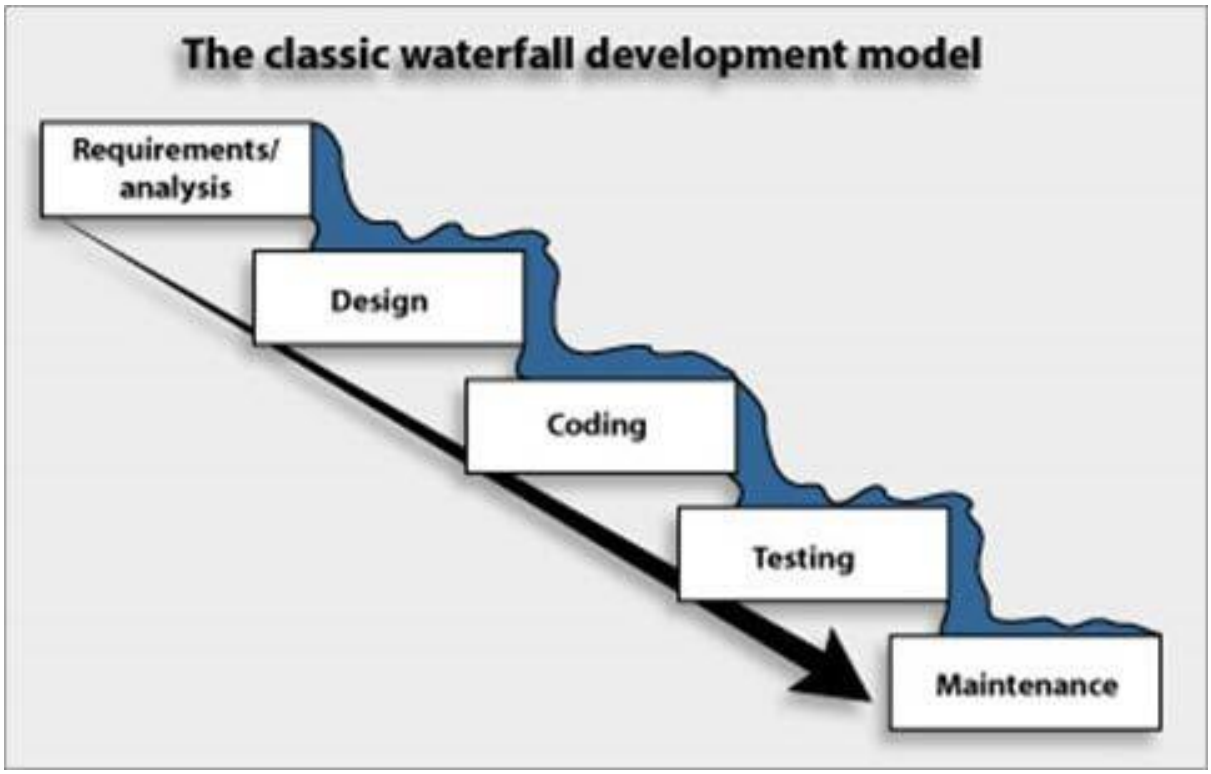
Ved at arbejde iterativt og inkrementelt kan man hurtigt identificere eventuelle problemer eller mangler, hvilket gør det muligt at rette op på dem tidligt i processen. Dette sparer tid og ressourcer da man ikke arbejder på noget som ikke er i overensstemmelse med virksomhedens behov og ønsker. Når man implementerer ændringer med det samme, kan man sikre en mere effektiv og kontinuerlig udviklingsproces.

Agile-metoden kræver et tæt samarbejde mellem virksomheden og udviklingsteamet. Dette indebærer regelmæssige møder, klare kommunikationskanaler og en åben dialog. Dette er afgørende for succes.

Agile-metoden er en nyere metode end den ældre Waterfall-metode. En meget simpel forklaring af Agile-metoden, man har en iterationen som bliver arbejdet og udviklet på indtil den er færdig og godkendt, man går ikke videre med andre dele af projektet før denne iteration er færdiggjort. Scrum

er små sprints, de defineres ved prædefineret tid, hvilket giver mulighed for at vide hvor lang tid projektet kommer til at tage inden næste sprint.

Waterfall-metoden:



Figur 2 - Waterfall-metoden

Waterfall-metoden er en sekventiel og lineær tilgang til projektledelse, hvor hvert trin i processen følger efter hinanden i en fast rækkefølge. Den er mere rigid og meget ligetil i forhold til Agile-metoden som er meget fleksibel og iterativ.

Når man gør brug af Waterfall-metoden er projektet opdelt i faster, der normalt inkluderer kravsspecifikation, systemdesign, implementering, test og drift. Hver fase skal udføres inden man kan gå videre til den næste fase og dette skaber en klar afhængighed imellem dem.

De standardfaser er følgende:

1. Kravspecifikation: Kravene til systemet defineres og dokumenteres detaljeret.
2. Systemdesign: Systemet designes på baggrund af de opstillet krav. Der udarbejdes tekniske specifikationer og arkitektur.
3. Implementering: Systemet udvikles og implementeres ud fra det udviklet design.
4. Test: Systemet testes grundigt og flittigt for at sikre at kravene bliver opfyldt og fungerer som ønsket.

5. Drift: System implementeres og tages i brug.

Waterfall-metoden er velegnet til projekter, hvor kravene er klart defineret og stabile. Hvor der er begrænset behov for ændringer undervejs. Den lineære natur gør det vanskeligt at tilpasse eventuelle ændringer eller feedback sent i projektet, dette kan føre til risiko for at levere et produkt der ikke opfylder kundens krav.

Værktøjer:

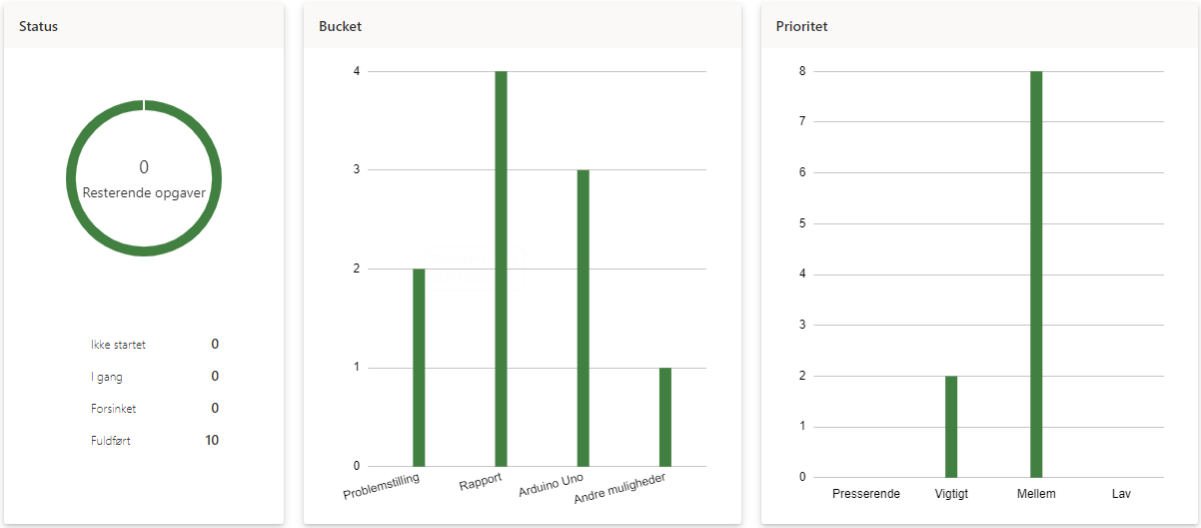
- Arduino Uno
- Arduino Uno IDE

Ansvar:

Denne rapport begyndte som en gruppe rapport, men på grund for uforudsete årsager er den en enkeltmandsgruppe nu. Da projektet startede, var der samarbejde som en gruppe og der blev stillet nogle klare ideer om hvordan projektet skulle udvikles, dette indebar bl.a. samarbejde med en virksomhed (Aarhus Eventpark), brugen af Microsoft planner for at tydeliggøre vores mål og gøre det ligetil at holde hinanden ajour på hvert projektdels fremskridt.

Som nævnt ovenover er dette nu et enmandsprojekt og dette indebærer at alt ansvar selvfølgelig ligger på den ene. Der er stadig blevet gjort brug af Microsoft planner for at tydeliggøre de forskellige mål der har været i projektet. Der har været en del tidsbegrænsning både i forhold til bruddet af gruppen, men også da der har været flere eksamener ind oveni arbejdstiden der har været sat af til projektet.

Normalt ville der have være udarbejdet en ansvarsliste som kunne påvise hvem der har haft ansvaret for hvad og om denne del af projektet også nåede at blive færdiggjort.



Figur 3- Microsoft Planner

Som kan ses ovenfor i Figur 3 ovenfor, så er Microsoft Planner blevet brugt til at skabe et overblik over projektets omfang samt give mulighed for at holde sig ajour i forhold til hvad der mangler og hvor meget tid der til at delen skal være færdiggjort.

Som det første blev der kigget på at få lavet en problemstilling, dette var dog lidt vanskeligt da den ikke måtte være megen til eller have samme mål som den resterende gruppes projekt. I planner blev der lavet to opgaver der fik to dages løbetid inden de skulle være færdiggjort.

Dette var at omformulere problemstillingen som den resterende gruppe forsat gør brug af og efterfølgende få godkendt den nye problemstilling. Dette var essentiel for at kunne starte projektet da det ligesom er med til at bestemme hvad hele projektet skal udvikles efter. Dette blev en tripwire alarm til selve hardball spillet og hvordan det kan have en indflydelse på at rykke spillets rammer.

Efterfølgende blev der lavet en brainstorm for ideer der kunne bruges til at udvikle et projekt der besvarer problemstillingen. Dette endte som sagt med en tripwire alarm, dertil kom delen hvor empiri skal indsamles og et produkt skal udvikles.

Da Arduino Uno blev valgt til at være hjertet i projektet, skulle der selvfølgelig samles information ind omkring enheden og hvad den er, gør og kan.

Dertil skulle der laves en mindre prototype som kan give en ide om hvordan det endelige produkt kan se ud og for at give en platform at arbejde ud fra. Dertil skal koden testes for at få de ønskede resultater, efterfølgende skal koden finpudses og blive lavet til en færdig del af projektet.

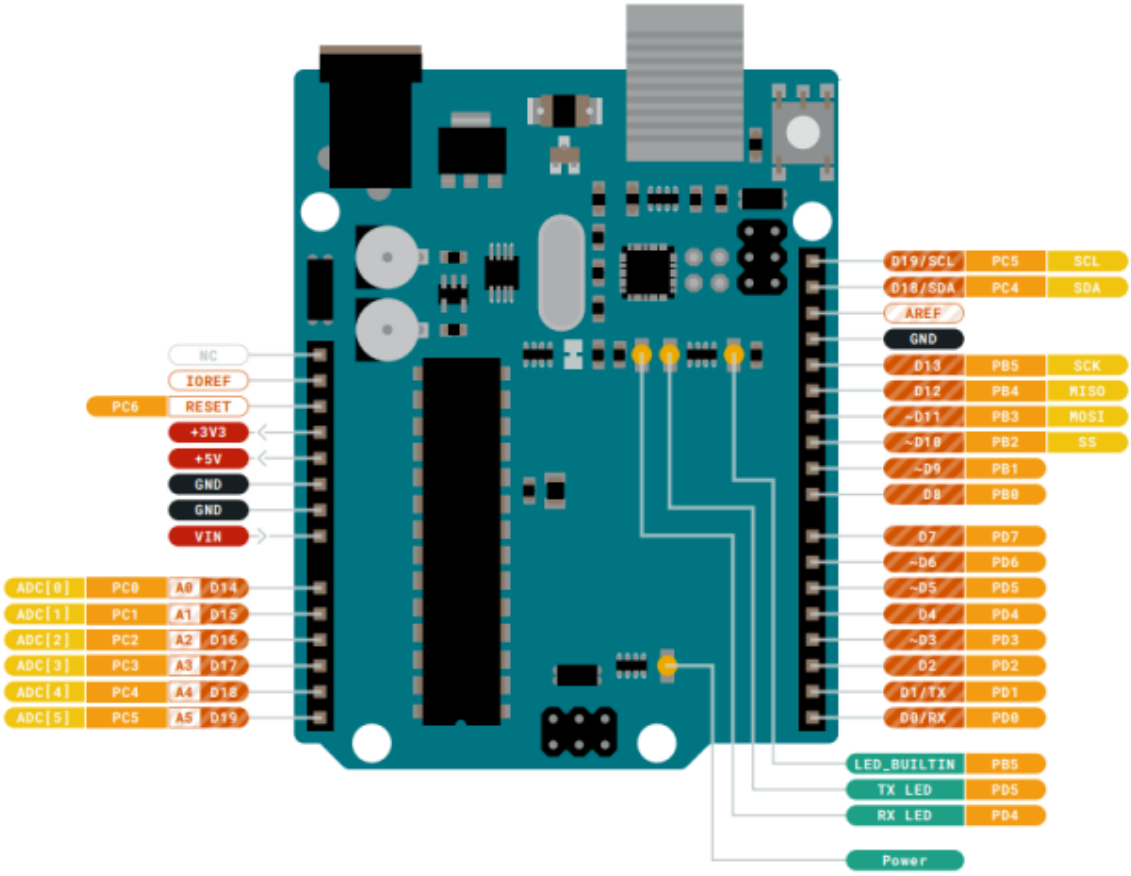
Sidst men ikke mindst er der selvfølgelig efter endt test og udviklet prototype. Så skal rapporten udarbejdes, den skal selvfølgelig være omkring både problemstillingen, hvordan målene er opnået og

hvordan man er nået dertil. De store hoveddele af rapporten består af den tunge tekst hvor komponenter og andet man har gjort brug bliver forklaret detaljeret hvilket gør at den næste der læser rapporten kan forstå og selv opsætte samme produkt.

Produktet og Procesteorien:

Som nævnt før i rapporten har der været forskellige ideer til hvad der kunne udvikles i forhold til at udvide hardball spillets rammer. Til en start har der været udtænkt en turret der kan skyde efter motion sensor eller manuel kontrol udviklet med Python. Dog faldt det på en tripwire alarm, som bliver lavet med en Arduino Uno og dertil tilhørende komponenter som samlet med kode skrevet på Arduino IDE i C++, giver en tripwirealarm der giver en lyd når en person passerer den. Dette blev besluttet da der tidsmæssigt ikke har været mulighed for at designe og udvikle en turret.

Arduino Uno:



Figur 4 - Arduino Uno

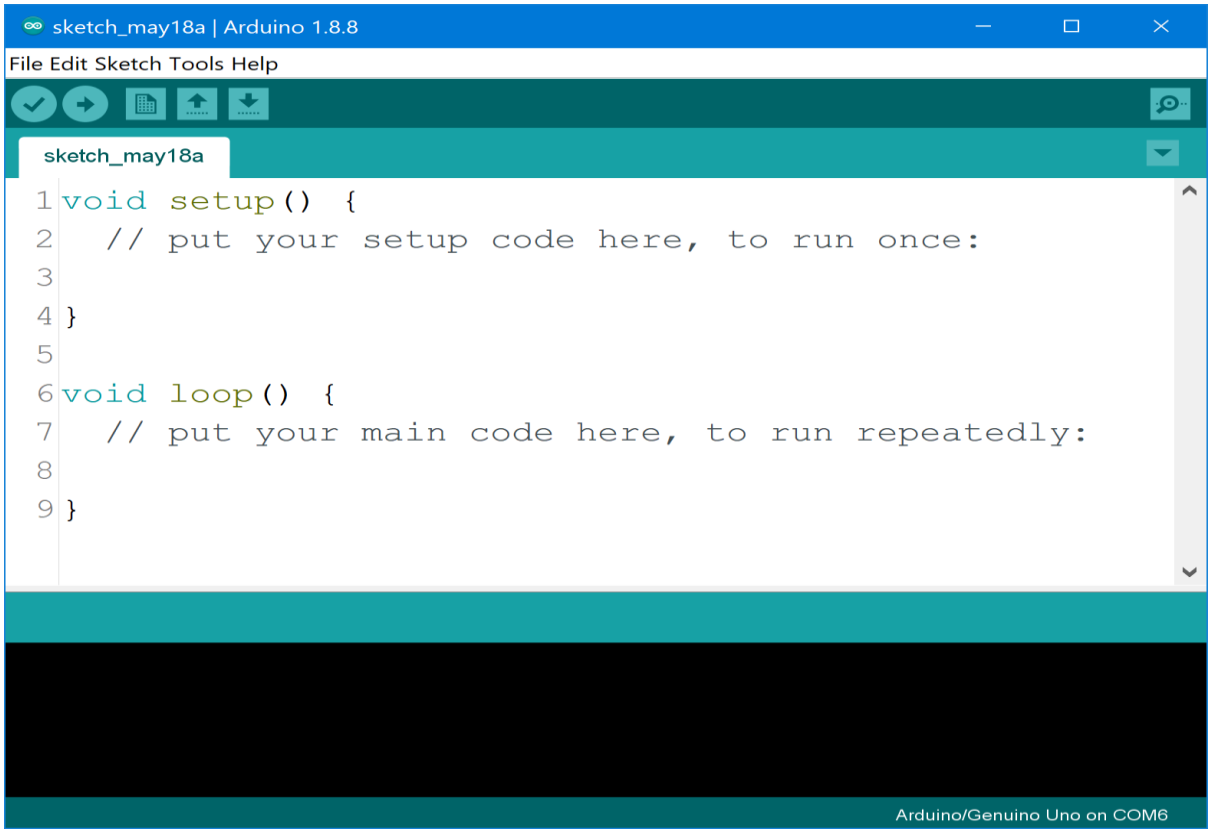
Arduino Uno er en mikrocontroller som er blevet et anerkendt værktøj til at lære elektronik og programmere i en bred vifte af projekter, den blev udviklet i 2005 med formået at skabe et brugervenligt og tilgængeligt system.

Arduino Uno er baseret på Atmel's ATMega328P mikrocontroller, som har en clockhastighed på 16 Mhz og en flashhukommelse på 32 KB. Den har 14 digitale input/output-pins hvoraf 6 kan bruges som PWM-udgange. 6 analoge input-pins, en usb-tilslutning til programmering og serielle kommunikation samt andre komponenter som strømforsyning og en reset-knap.

Et af de mest bemærkelsesværdige funktioner ved Arduino Uno er dens åbne kildekolde og den store mængde af tilgængelige ressourcer og dokumentation. Arduino Uno er blevet brugt i et bredt spektrum af projekter, b.la. robotter, hjemmeautomatisering, og mere.

En Arduino Uno er til dette projekt perfekt, der er en masse dokumentation at få inspiration fra, den er nem at gå til og kan uden videre opfylde de krav der er til en opsætning af en tripwire alarm. Den kan også kodes og derefter bruges uden forbindelse til den, kun strøm er en nødvendighed, så på den front er den perfekt transportabel i dette tilfælde.

Arduino IDE:



Figur 5 - Arduino IDE

Arduino IDE (Integrated Development Environment) er softwareudviklingsplatform til Arduino-platformen. Den giver mulighed for at skrive, kompilere og uploade programmer til mikrocontrolleren.

Det er en dejlig brugervenlig platform, den kan bruges af både nybegyndere og øvede brugere. Den er tilgængelig på flere OS's, bl.a. Windows, Mac OS X og Linux, hvilket øger den tilgængelighed.

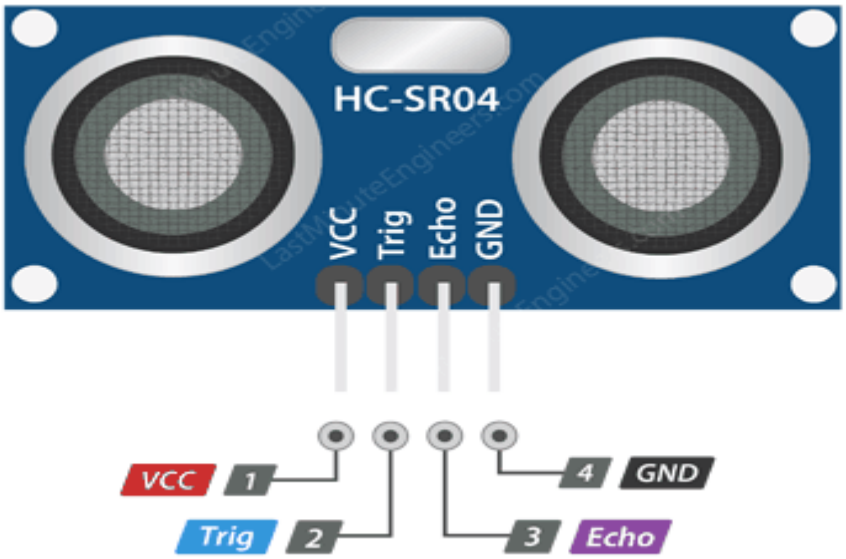
Den har en enkelt og overskuelig brugergrænseflade, der består af et tekstredigeringvindue, en værktøjslinje og et resultatvindue. Brugergrænseoverfladen giver enkel adgang til de nødvendige værktøjer og funktioner som skal bruges til at udvikle programmer.

I tekstredigeringsvinduet kan koden blive skrevet, den understøtter selvfølgelig syntaxhøjdepunkter, automatisk indrykning og auto-fuldførelse hvilket gør det meget nemmere at skrive og finde fejl i sin kode.

Når man har skrevet koden kan man bruge værktøjslinjen til at kompilere og uploade sit program til mikrocontrolleren. Der er også muligheder for at vælge den korrekt Arduino-boardmodel, port indstillinger når man skal have adgang, kompilering og upload-knapper.

Dette software er givent nødvendigt for projektet da det er den nemmeste og ligetil tilgang til at få skrevet den kode og lavet det program der skal til for at få projektet ud i verdenen.

HC-SR04:



Figur 6 - HC-SR04 Ultralydssensor

HC-SR04 er et af de komponenter der skal bruges til at lave tripwire alarmen, den har en meget vigtig funktion da det er den der skal registrere når en person går forbi den, hvilket resulterer i en alarm lyd. HC-SR04 er en ultralydssensor, den kan bruges til afstandsmåling eller i det her tilfælde hindringsdetektion. Den er enkel, præcis og pålidelig, den kommer til en overkommelig pris ligesom så meget andet til Arduino gør.

Den består af to primære komponenter, en ultralydsender og en ultralydsmodtager. Ultralydssenderen udsender ultralydsignaler i form af korte pulser og ultralydsmodtageren opfanger disse og registrerer dem. Funktionsmåden for HC-SR04-sensoren er baseret på tidsforsinkelsen mellem afsendelsen af ultralydsignalet og modtagelsen af det reflekterede signal. Ved at måle denne tidsforsinkelse kan sensoren bestemme afstanden til et objekt foran sensoren.

Typisk har HC-SR04-sensoren et måleområde på omkring 2 cm til 400 cm, dette gør den velegnet til korte og lange afstandsmålinger. Selv i forskellige miljøer er den forholdsvis nøjagtig og stabil.

Buzzer 5v Active:

Figur 7 - Aktiv Buzzer 5v

En buzzer er en elektromekanisk enhed der bruges til at generere lyde eller i dette tilfælde et advarselssignal. Den er anderledes fra en passiv buzzer ved at den indeholder en indbygget oscillerende kreds, som producerer lyd når der påføres en spænding.

En aktiv buzzer består af en piezoelektrisk keramisk disk, en intern oscillator og en forstærker. Når en vekselstrøm eller pulserende spænding påføres buzzeren, vibrerer den piezoelektriske disk hvilket skaber lydbølger og producerer en hørbar tone.

Man styrer buzzeren ved at påføre vekselspænding med en passende frekvens og varighed. Ved at ændre frekvensen af den påførte spænding kan man producere forskellige toner og lyde. Buzzeren kan også aktiveres og deaktiveres ved at kontrollere til og frakoblingen af strømmen.

Aktiv 5v buzzer er praktiske at bruge når man arbejder med en mikrocontroller som en Arduino Uno da den drives direkte fra 5V-strømforsyningen uden behov for ekstra komponenter såsom transistorer eller relæer. Der skal to ledninger til at tilslutte den, en til VCC (+5V) og en til Ground (GND).

Aktive buzzere har forskellige lydstyrker og tonearter afhængigt af deres design og specifikationer. Nogle buzzere kan have en fast tone, mens andre kan være programmerbare til at producere

forskellige toner eller melodier. I dette tilfælde er en høj klar tone alt der er brug for at give tripwire alarmen et af dens vigtige funktioner, nemlig alarm gennem lyd.

Mulig pris for prototypen:

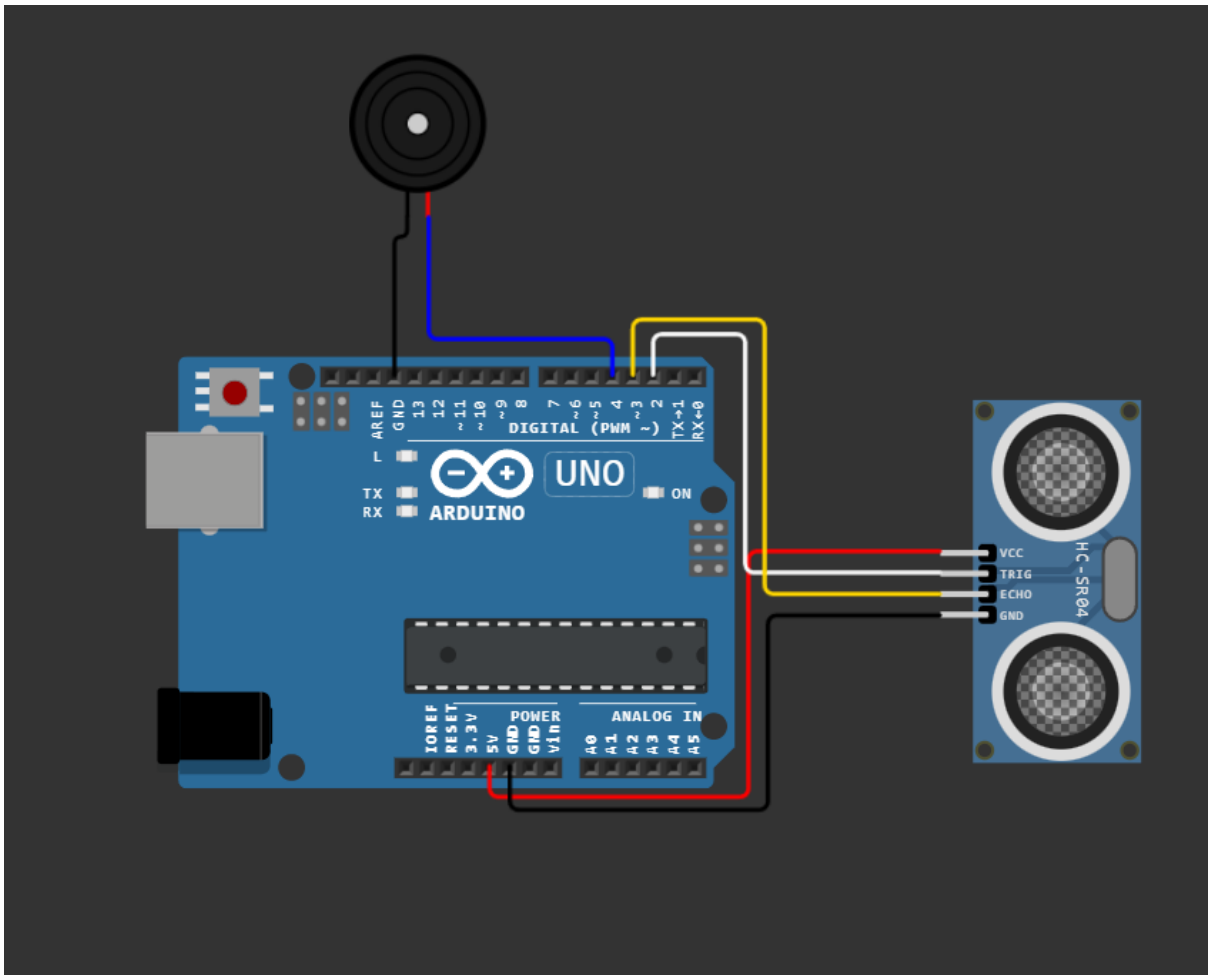
Heldigvis når det kommer til at skulle købe Arduino Uno og komponenter til den hjem, så er det meget lige til, der eksisterer mange hjemmesider der sælger både enkelte dele og pakker. Priserne i forhold til hvad du kan lave, er forholdsvis lave, hvilket er et stort plus hvis man skal udvikle mere end et produkt. I det her tilfælde er der kun en prototype, men havde det nu skulle være et færdig udviklet produkt med et budget, så kunne man gøre det billigt alt taget i betragtning.

Komponent	Pris	Hjemmeside
Arduino Uno	178.75 kr.	Arduino Store[2]
Aktiv Buzzer 5V	10,27 kr.	Arduino Store[3]
HC-SR04	48,75 kr.	Let-elektronik.dk[4]
Kabler	119,95 kr.	3deksperten.dk[5]
Strømforsyning	65.00 kr.	Ardustore.dk[6]
USB-kabel	19,00 kr.	Av-cables.dk[7]
I alt	441,72 kr.	

Som kan ses i opslaget ovenfor kan man slippe afsted med en tripwire alarm for under 500 kr. pr enhed, hvilket igen tydeliggør hvor billigt man kan slippe afsted med at udvikle sådan et produkt. Det er jo i sidste ende heller ikke det mest avancerede projekt, hvis man har tiden, viljen og lysten, kunne man nemt lave en i sin egen fritid.

[2] https://store.arduino.cc/products/arduino-uno-rev3
[3] https://store.arduino.cc/products/grove-buzzer-piezo?queryID=33029e5b8c24f50593158bc438dda11d&_gl=1*1bzzn1c*_ga*ODQxNTUyMDc2LjE2ODQxNDkzNTY.*_ga_NEXN8H46L5*MTY4NDkyMDY3MS44LjEuMTY4NDkyMDcwMi4wLjAuMA..
[4] https://let-elektronik.dk/ultrasonic-distance-sensor-hc-sr04?gclid=CjwKCAjw67ajBhAVEiwA2g_jEBoGodJ8zpnSaw0b-dGYzVcfbAW3ZG1lXDYlYJ6LitEZRQwqBJn_VRoC6ngQAvD_BwE
[5] https://3deksperten.dk/products/635pcs-dupont-connector-male-female-2-54mm-with-1-5m-10pin-dupont-cable?variant=43307622662364&gclid=CjwKCAjw67ajBhAVEiwA2g_jELx0NQ7TAT_1xcGu05WOjgRwBnHtEz2mjSFQdIDTrOJcl8cq9OCfWxoCTZ4QAvD_BwE
[6] https://ardustore.dk/produkt/power-supply-adapter-dc-5v-2a?gclid=CjwKCAjw67ajBhAVEiwA2g_jECkTdfypU-pX86b_28fwQCN2EK3H-oVpHmmmWP1v1HCtTgOQmcrTFRoCI1MQAvD_BwE
[7] https://www.av-cables.dk/usb-a-usb-b-kabel/usb-kabel-2-0-usb-a-han-usb-b-han-0-25-m.html?utm_source=google&utm_medium=organicshopping&utm_campaign=googleshoppingorganic&gclid=CjwKCAjw67ajBhAVEiwA2g_jEFcU2xXFXSTLkSieRbvPaugSp5j3VCeuqc1lVOt79xth197bQO0r_hoCuMQQAvD_BwE

Implementering og Analyse:



Dette er et simpelt diagram der viser opsætningen af tripwire alarmen, vi har vores Arduino Uno, vores buzzer og vores ultralydssensor.

Koden har jeg fundet inspiration gennem open-source projekter som andre har lavet både med formål der ligner eller irrelevante i forhold til projektet. Men det giver en god indsigt i hvordan man kan kode og hvad man kan gøre med de gældende komponenter.

```
const int triggerPin = 2;     // Pin til at sende trigger-signal til HC-SR04
const int echoPin = 3;        // Pin til at modtage echo-signal fra HC-SR04
const int speakerPin = 4;     // Pin til tilslutning af højtaleren
```

Figur 8 - Definerer pins

Først skal pins defineres, det er nødvendigt ellers falder hele projektet lidt til jorden. Vi har "triggerPin" det er en digital pin som vi bruger til at sende et trigger-signal til vores HC-SR04-sensor, "echoPin" er den digitale pin vi bruger til at modtage vores echo-signal fra sensoren. "speakerPin" er en digital pin vi bruger til at forbinde vores buzzer.

```
int sensorThreshold = 50;     // Tærskelafstand for at aktivere tripwire (i centimeter)
boolean tripwireActive = false;  // Variabel til at spore tripwire-tilstanden
```

Figur 9 - Definerer afstand

"SensorThreshold" bliver defineret som vores tærskelafstand, når den brydes, vil den udløse tripwiren, jeg har valgt at måle i centimeter.

```
void setup() {
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(speakerPin, OUTPUT);
  Serial.begin(9600);
}
```

Figur 10 - Konfigurerer OUTPUT og INPUT

I mit void setup konfigurerer jeg mine pins som henholdvis OUTPUT og INPUT, derudover tilføjede jeg en "Serial.begin(9600);", altså en serial-kommunkation med en baudrate på 9600.

```
void loop() {
  long duration, distance;

  // Sender ultralydssignal
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  // Måler tiden for ultralydssignalet til at returnere
  duration = pulseIn(echoPin, HIGH);

  // Beregner afstanden baseret på tiden
  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > 0 && distance < sensorThreshold && !tripwireActive) {
    tripwireActive = true;
    playSound(1000, 500);  // Afspil en lyd med 1 kHz frekvens i 500 ms
    Serial.println("Tripwire activated!");
  }

  if (distance >= sensorThreshold) {
    tripwireActive = false;
  }
}
```

Figur 11 - Opsætning af void loop

I min loopfunktion som kører kontinuerligt og står for at udsende et ultralydssignal ved at sætte ”triggerPin” til LOW i 2 mikrosekunder, derefter går den til HIGH i 10 mikrosekunder og til allersidst tilbage til LOW igen. Det er denne funktion der udløser sensoren til at sende en ultralydspuls.

”pulseIn()” funktionen bruger vi til at måle tiden det tager for echo-signalet at returnere tilbage, tiden bliver gemt i variablen ”duration”. Afstanden beregner vi ved at gange ”duration” med konstanten 0,034 og derefter dividere vi den med 2. dette er en omdannelsesfaktor fra tid til afstand, da lyden rejser med en hastighed på ca. 34cm/ms.

Afstandsværdien i cm udskrives til Serial Monitor ved at vi gør brug af ”Serial.print” og ”Serial.printLn” funktioner.

Hvis vores værdi fra ultralydssensoren er mellem 0 og vores satte "sensorThreshold", og vores tripwire ikke allerede er aktiveret, så udløses vores tripwire. Vi kan markere dette ved at sætte "tripwireActive" til true og afspille vores alarm lyd ved hjælp af "playSound()" funktionen. Der bliver også udskrevet en bekræftelse af tripwiren aktiveres til vores Serial Monitor. Hvis vores værdi er større eller lig med "sensorThreshold" og tripwiren er aktiveret sættes "tripwireActive" til false for at nulstille tripwire-tilstanden.

Vores "playSound" funktion bruges til at kunne afspille vores alarm lyd på højtaleren. Den tager imod en frekvens og varighed som dens argumenter. Ved hjælp af en simpel til-/fra-metode generes en firkantbølge med den angivne frekvens og varighed ved at skifte højtalerens tilstand med passende pauser.

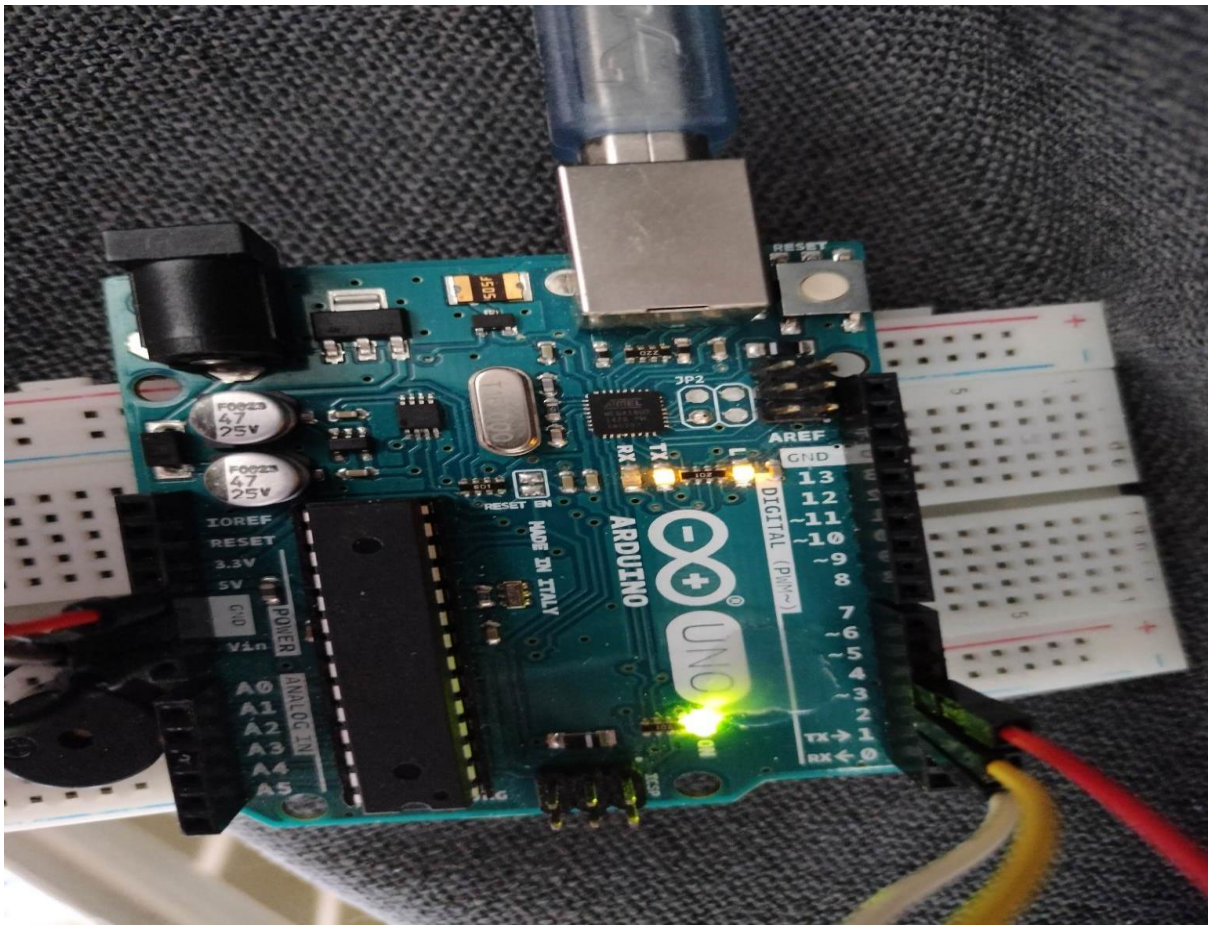
```
// Funktion til at afspille en lyd på højtaleren
void playSound(unsigned int frequency, unsigned long duration) {
  unsigned long period = 1000000 / frequency;  // Beregning af perioden for den ønskede frekvens
  unsigned long halfPeriod = period / 2;       // Halvperiode til toggling af højtalerens tilstand
  unsigned long numCycles = frequency * duration / 1000;  // Antal cyklusser baseret på ønsket varighed

  for (unsigned long i = 0; i < numCycles; i++) {
    digitalWrite(speakerPin, HIGH);    // Skifter højtalerens tilstand til HIGH
    delayMicroseconds(halfPeriod);     // Venter i halvdelen af perioden
    digitalWrite(speakerPin, LOW);     // Skifter højtalerens tilstand til LOW
    delayMicroseconds(halfPeriod);     // Venter i halvdelen af perioden
  }
}
```

Figur 12 - Funktion "playSound"

Test:

Først og fremmest før testning starter sikrer jeg mig at hver komponent jeg skal bruge, bliver testet grundigt for at sikre at komponenten fungerer som den skal. Dette kan gøre ved at bruge de allerede vel dokumenteret sketches som er tilgængelige når man bruger Arduino IDE.

Som det første skal Arduino Uno testet, man kan kontrollere den ved at holde øje med om der kommer lys på boardet når der bliver sat strøm til den, hvis du får dine grønne lys, er du good to go.



Figur 13- Arduino Uno LED lys

Der er blevet brugt en simpel sketch til at verificere at Arduino Uno fungerer som den skal:

```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Figur 14 - Arduino Uno LED Blink test

Jeg gør brug af en simpel blink sketch som tester Arduino Uno's indbyggede LED, på den måde kan jeg sikre mig at den kan bruges til at tage imod og gøre brug af de programmer/kode som jeg fodrer den med.

Den næste komponent jeg testede igennem, var HC-SR04, her gælder det samme princip igen, jeg kobler den til Arduino Uno og finder en sketch der kan give mig en indikation om at HC-SR04 fungerer korrekt eller efter hvad jeg skal bruge den til.

```
/*
  Ultrasonic Sensor HC-SR04 and Arduino Tutorial

  by Dejan Nedelkovski,
  www.HowToMechatronics.com

*/
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

Figur 15 - HC-SR04 test

I dette tilfælde fandt jeg en sketch online som jeg kunne gøre brug af[8], som kan ses så vil jeg få et print i min Serial Monitor som jeg kan gøre brug af for at se hvad HC-SR04 faktisk måler i realtid. Dette bruger jeg så til at vurdere om den giver mig rigtige målinger eller om den er defekt.

[8] https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Den sidste komponent der skal testet er min Buzzer Active 5V, jeg følger det samme princip igen, jeg finder en sketch der giver mig mulighed for at lave en simpel test og få en indikation om hvor god komponent fungerer.

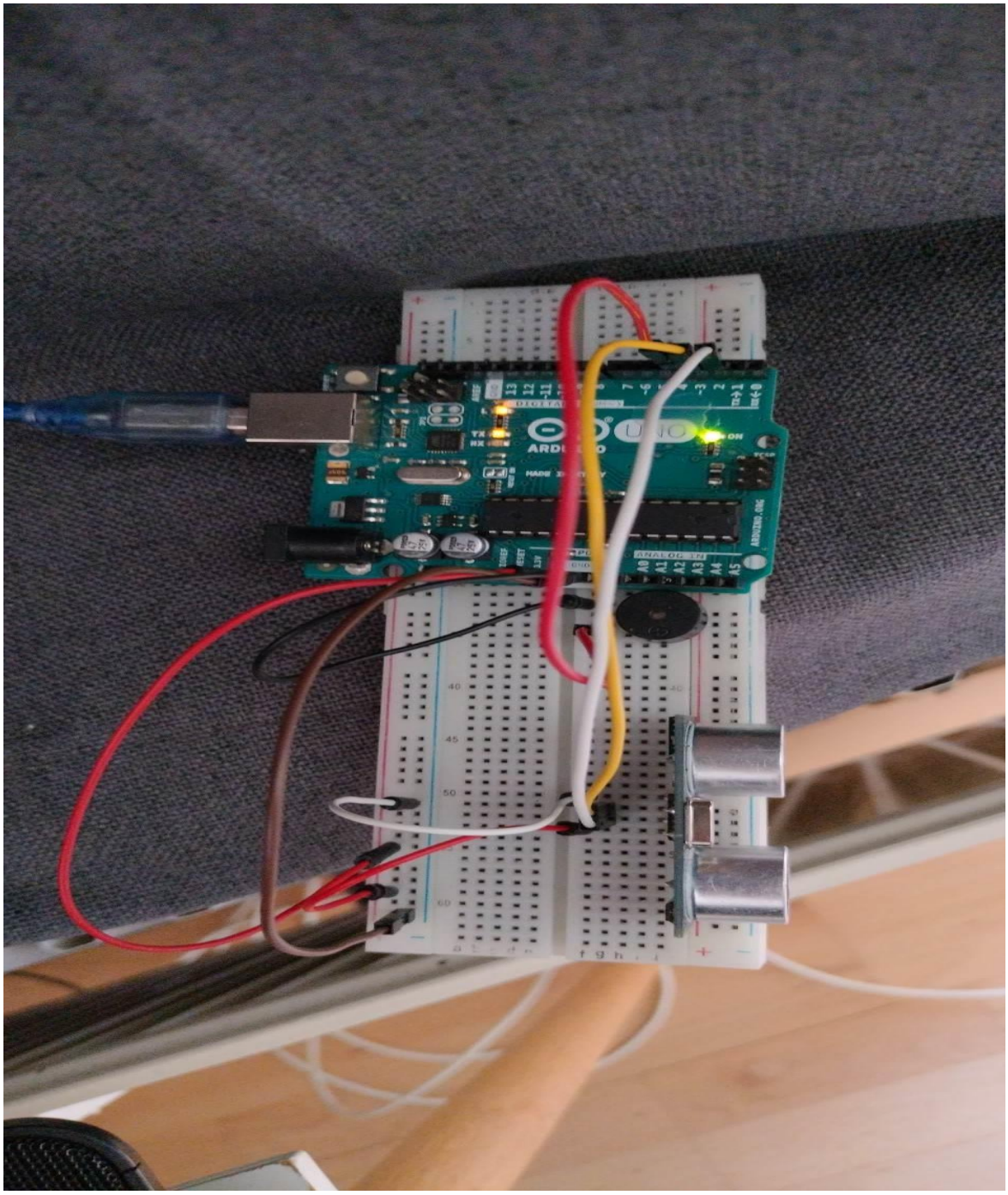
```
/* Arduino tutorial - Buzzer / Piezo Speaker
   More info and circuit: http://www.ardumotive.com/how-to-use-a-buzzer-en.html
   Dev: Michalis Vasilakis // Date: 9/6/2015 // www.ardumotive.com */

const int buzzer = 9; //buzzer to arduino pin 9

void setup(){
  pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output
}

void loop(){
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);        // ...for 1 sec
  noTone(buzzer);     // Stop sound...
  delay(1000);        // ...for 1sec
}
```

Figur 16 - Buzzer Test

I dette tilfælde fandt jeg en sketch online igen[9], og når jeg kører sketchen med min buzzer tilsluttet, får jeg min ønsket lyd.

Nu har jeg testet alle mine komponenter igennem og kan bekræfte at hver del for sig virker. Nu skal det hele kobles sammen og der skal laves nogle test i forhold til om de virker alle samlet.

[9] https://www.ardumotive.com/how-to-use-a-buzzer-en.html



Figur 17 - Tripwire Alarm

Når nu alle delene er samlet, kan jeg gøre brug af min egen kode som vist tidligere i rapporten. Dog skal der være nogle kriterier der skal være indenfor hvad jeg ønsker for min tripwire alarm.

Der skal laves en afstandstest, det kan jeg gøre ved at placere sensoren et godt sted, hvor jeg har mulighed for at passere den, det kan jeg så gøre ved forskellige afstande for at se hvornår den giver alarm.

Der er også blevet testet for følsomhed, den skal helst kunne samle en hurtig bevægelse op, så det kan man teste ved at lave hurtige bevægelser med en hånd foran sensoren eller løbe forbi den.

Der skal også testet for dens pålidelighed, det er ikke nødvendigvis at den virker fordi du kan aktivere den 3 gange, den skal helst registrere hver gang, ligegyldig hvilken form for bevægelse der bliver lavet inden for sensoren rækkevidde.

Der skal også laves en lyd test for at teste om lyden er som ønsket. Jeg har skruet ned for den i det her tilfælde da man skal kunne holde det ud, men hvis den skulle bruges på en bane, så ville en høj trælstone være perfekt, så man skal huske at prøve sig frem med forskellige lydniveauer, det ville være bedst at få mulighed for at teste det på selve banen for at høre hvordan lyden folder sig ud der, i forhold til hjemme i en stue eller et klasseværelse.

Sidst men ikke mindst, hvis man kører Arduino Uno på batterier for at give spillerne muligheden for at tage den med rundt, så skal man teste batteriernes levetid, simpel løsning ville være at slå den til og dokumentere hvornår den går ud og batterierne skal udskiftes.

Perspektiver og Refleksioner:

Samarbejdet med virksomheden Aarhus Eventpark startede med et intromøde med lederen for Aarhus Hardball som hører under Aarhus Eventpark. Dette møde blev holdt for at etablere hvad for et produkt virksomheden kunne ønske sig at få udviklet. Under mødet blev der kastet mange ideer rundt, men i forhold til hvad der skulle laves, var fuldt ud op til os der skal lave projektet. Oprindeligt var det et gruppeprojekt hvor der var lavet en problemstilling ud fra ideen om at udvikle en turret til at tilføje til spillets rammer. Da der desværre kom et dødsfald i familien, måtte der tages forbehold for dette, desværre var resultatet at gruppen ikke var villige til at forsætte et arbejde sammen og dette fører så til at projektet bliver til et enmandsprojekt med en ny problemstilling og et nyt produkt at udvikle. Alt dette med meget begrænset tid, både på grund af forpligtelser til skole, men også tidsmæssigt da alt arbejde inden gruppen blev splittet ikke kan bruges fremadrettet. Derfor måtte jeg tilbage til tegnebordet og starte fra bunden igen.

Projektet begyndte med at skulle udvikles i Python med en Raspberry Pi, der skulle udvikles en manuel turret der kunne styres både med mus og et joystick. Dog grundet tidsmæssige årsager ville det ikke være muligt at udvikle et ordentligt produkt, hverken teoretisk eller færdig med en prototype. Efterfølgende blev det til en form for alarm, dette blev omtalt under det første intromøde og dertil kom ideen om at udvikle en simpel tripwire alarm som kan have enormt funktionalitet for hvordan spillets rammer og regler pludselig kan blive ændret.

Dette blev først tænkt ind som en mulighed for at forsat gøre brug af en Raspberry Pi, men det viste sig efter indsamling af empiri og data, at det ville være mere simpel og ligetil at gøre brug af en Arduino Uno med tilsvarende komponenter. Der er masser af dokumention at få inspiration fra, og skolen havde komponenterne ved hånden. Dette ville gøre hele processen i at få udviklet en ide indtil en prototype langt hurtigere og mere stabil.

Anbefalinger:

For at opstarte et projekt med en god start, er det enormt vigtigt at sætte tiden til at lave en gennemgribende og ihærdig indsamling af data og undersøgelser for at kunne lave et scope på og et muligt design som er tilsvarende ønsket fra virksomheden man udvikler et produkt for. Efterfølgende kan man skabe sig et overblik over hvad projektet kommer til at indebærer og hvordan man bedst muligt kan sammensætte det endelige produkt efter man har arbejdet med sine delmål.

I forhold til projekter med et produkt der skal udvikles, så er det altid vigtigt at man kan få adgang til de nødvendige ressourcer, dette vil hovedsageligt være et budget til at skaffe personale til at arbejde på projektet, men også til at skaffe nødvendige ressourcer såsom hardware, software og hvad der nu ellers er nødvendigt for at projektet får succes.

I dette tilfælde er projektet så småt at selv en enkelt person ville kunne udføre det. Dog er der hele design delen af projektet som muligvis kunne blive outsources til et andet firma eller freelancer. Det ville heller ikke være et kosteligt projekt da Arduino Uno og mange af dens komponenter er forholdsvis billige at anskaffe, dette kan ses længere oppe i rapporten hvor der er et prisanslag på hvad det kunne koste at bygge med kun hardware komponenter en tripwire alarm.

Konklusion:

I dette projekt om udviklingen af en tripwire alarm som skal bruges til hardball spillet har været et lærerigt og udfordrende projekt. Til start i projektet var der en klar problemstilling og en vision om at implementere en fungerende turret ind i spillet. Da et gruppebrud forekom, blev hele projektet lavet om, der skulle en ny problemstilling og et nyt produkt til. Dette blev til en tripwire alarm.

Gennem forskning og analyse af data fundet online, blev der fundet frem til nøglekomponenter som Arduino Uno, HC-SR04 ultralydssensor og en aktiv buzzer, som er afgørende for alarmens funktion. Arduino IDE blev brugt som udviklingsmiljøet til programmering og styring af komponenterne, projektet blev gennemført med en Agile-metode, der tillod fleksibilitet og tilpasningsevne undervejs, hvilket var meget nødvendigt da der var et stort tidspres.

Der er blevet udarbejdet en detaljeret beskrivelse af hver komponents funktionalitet, bl.a. deres tekniske specifikationer og tilslutning. Implementeringen af tripwire alarmen er blevet simuleret og beskrevet teoretisk, da tiden og ressourcerne ikke tillod et færdigt produkt og kun en simpel prototype.

Projektet har givet nogle vigtige perspektiveringer og refleksioner. Der er kommet en vished om hvor vigtigt det er at arbejde inden for tidsrammer og tilpasse sig ændringer og udfordringer undervejs. Planlægning og prioritering af ressourcer er afgørende for succes.

Selvom projektet ikke nåede sit oprindelige mål om at producere en funktionelt færdig produceret model af en tripwire alarm, så har det alligevel bidraget med vigtig viden og erfaring inden for teknologien bag konstruktionen og implementeringen.

Samlet set har dette projekt været en vigtig læringsproces. Selvom det endelige resultat måske ikke opfyldte alle forventninger, har det alligevel lagt et fundament for fremtidige projekter og bidraget til en dybere forståelse af vigtigheden af effektivt projektarbejde.

Bibliografi:

1. https://da.myservername.com/agile-vs-waterfall-which-is-best-methodology
2. https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf
3. https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet
4. https://components101.com/misc/buzzer-pinout-working-datasheet
5. https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics
6. https://da.wikipedia.org/wiki/Hardball
7. https://tasks.office.com/itcn.onmicrosoft.com/da-DK/Home/Planner/

Appendix:

1. https://da.wikipedia.org/wiki/Hardball
2. https://store.arduino.cc/products/arduino-uno-rev3
3. https://store.arduino.cc/products/grove-buzzer-piezo?queryID=33029e5b8c24f50593158bc438dda11d&_gl=1*1bzzn1c*_ga*ODQxNTUyMDc2LjE2ODQxNDkzNTY.*_ga_NEXN8H46L5*MTY4NDkyMDY3MS44LjEuMTY4NDkyMDcwMi4wLjAuMA..
4. https://let-elektronik.dk/ultrasonic-distance-sensor-hc-sr04?gclid=CjwKCAjw67ajBhAVEiwA2g_jEBoGodJ8zpnSaw0b-dGYzVcfbAW3ZG1lXDYlYJ6LitEZRQwqBJn_VRoC6ngQAvD_BwE
5. https://3deksperten.dk/products/635pcs-dupont-connector-male-female-2-54mm-with-1-5m-10pin-dupont-cable?variant=43307622662364&gclid=CjwKCAjw67ajBhAVEiwA2g_jELx0NQ7TAT_1xcGu05WOjgRwBnHtEz2mjSFQdIDTrOJcl8cq9OCfWxoCTZ4QAvD_BwE
6. https://ardustore.dk/produkt/power-supply-adapter-dc-5v-2a?gclid=CjwKCAjw67ajBhAVEiwA2g_jECkTdfypU-pX86b_28fwQCN2EK3H-oVpHmmmWP1v1HCtTgOQmcrTFRoCI1MQAvD_BwE
7. https://www.av-cables.dk/usb-a-usb-b-kabel/usb-kabel-2-0-usb-a-han-usb-b-han-0-25-m.html?utm_source=google&utm_medium=organicshopping&utm_campaign=googleshoppingorganic&gclid=CjwKCAjw67ajBhAVEiwA2g_jEFcU2xXFXSTLkSieRbvPaugSp5j3VCeuqc1lVOt79xth197bQO0r_hoCuMQQAvD_BwE