



LUNDS TEKNISKA HÖGSKOLA
Lunds universitet

EITF12: Whac_a_Mole

Projektrapport

Anna Wahlund, Lova Sedigh & Sofie Gradén
Grupp 15

Handledare: Christoffer Cederberg

2022-05-19

Innehållsförteckning

1. Inledning

1.1 Bakgrund

1.2 Syfte

2. Kravspecifikation

3. Hårdvara

3.1 Komponenter

4. Mjukvara

5. Metod

5.1 Planering

5.2 Konstruktion

5.3 Programmering

6. Resultat

7. Diskussion

8. Referenser

9. Bilaga

1. Inledning

1.1 Bakgrund

I kursen Digitala Projekt, EITF12, ges studenter möjligheten att konstruera en hårdvara och samtidigt implementera tillhörande mjukvara. I den här rapporten behandlas ett sådant projekt där ett spel, Whac_a_Mole, konstruerats. Genom spelet får två användare möjligheten att tävla mot varandra i reaktionsförmåga.

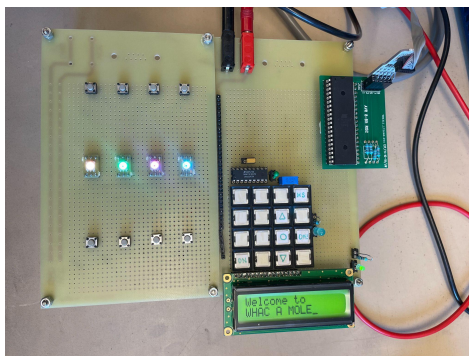
1.2 Syfte

Syftet med projektet, och kursen i helhet, är att få förståelse för industriellt utvecklingsarbete genom att utveckla en prototyp för vidareutveckling med nödvändig dokumentation.

2. Kravspecifikation

- Innan spelets start visar displayen hur spelaren ska påbörja spelet. Detta ska göras genom att trycka på ON-knappen.
- Under en spelomgång tänds lampor 15 gånger. Det finns fyra lampor och ordningen de tänds i slumpas. Tiden mellan att lamporna tänds slumpas också mellan en till fem sekunder.
- Varje gång en spelare trycker på knappen som motsvarar den lampan som tänds, slocknar lampan. Spelet jämför de två spelarnas knapptryck och ger den som tryckt snabbast poäng.
- Efter att lamporna tänts 15 gånger presenterar spelet vinnaren av omgången och därefter poängställningen.
- Spelet sparar det bästa resultatet. Körs spelet och en spelare får ett resultat som är högre än det nuvarande rekordet, får spelaren möjlighet att skriva in sitt namn med hjälp av knappsatsen. Spelet uppdaterar sitt minne med det nya namnet och den nya poängen. Rekordet sparas i EEPROM-minnet.
- Fås ett nytt rekord presenteras det på displayen efter att namnet skrivits in. Rekordet kan även skrivas ut med hjälp av knappen HS.

3. Hårdvara



Figur 1. Bild på den slutgiltiga hårdvaran.

3.1 Komponenter

- Processor

I det här projektet används en ATmega16-processor, vilket är en 8-bitars mikrokontroller baserad på AVR-förstärkt RISC-design. [1] I Atmel Studio 7.0 programmeras processorn i programspråket C, och mjukvaran laddas upp i processorn via en JTAG.

- Display

Displayen som används i Whac_a_Mole är en bakgrundsbelyst LCD display med 2 rader om 16 tecken vardera.

- Lysdioder

Till spelet används seriekopplade adresserbara RGB lysdioder.

- Knappar

Till spelet används åtta knappar.

- Knappsats

En knappsats med 16 knappar används.

- Key Encoder

En key encoder används för att processorn ska kunna tolka vad som görs på tangentbordet. Den key encoder som använts har 18 portar.

Se *bilaga 1* för en illustration av det kopplingsschema som hårdvaran konstruerats utefter.

4. Mjukvara

- Adobe Illustrator 2019 – Framtagning av kopplingsschema.
- Atmel Studio 7.0 – Programmering av mjukvaran till Whac_a_Mole.

5. Metod

5.1 Planering

Inledningsvis spånades idéer kring vad projektet skulle bestå av, givet kursens och utvecklarnas förutsättningar. Det landade snabbt i att ett spel skulle utvecklas, och därefter kom idén att utveckla Whac_a_Mole. Spelets initiala kravspecifikation togs sedan fram och förbättrades efter konsultation med handledare. Därefter beslutades det om vilka komponenter hårdvaran till Whac_a_Mole skulle vara uppbyggd av.

5.2 Konstruktion

När komponenterna till hårdvaran var valda, påbörjades uppbyggnaden av hårdvaran. Inledningsvis sattes en röd och en svart kontakt på kretskortet för att kunna koppla ström och jord till spelets processor. Därefter löddes olika komponenter, såsom processor, kondensatorer, potentiometer och pins, på kretskortet. Lödningen utfördes med hjälp av lödtråd och en magnstat-lödkolv. Med hjälp av tråd och en virpistol, kopplades sedan komponenterna till VCC och GND. Därefter löddes display, knappsats, knappar och lysdioder fast på motsvarande sätt. Motstånd kopplades till knapparna. Då spelet kräver en knappsats, åtta knappar och fyra lysdioder, som alla kopplas till processorn, används seriekopplade adresserbara RGB lysdioder för att antalet pins på processorn ska räcka till. För att koppla samma komponenterna korrekt användes komponenternas datablad.

Under uppbyggnadens gång testades de olika komponenterna för att kontrollera att kopplingarna utförts korrekt. Det här gjordes genom att koppla in temporära lysdioder. Lysdiodernas funktion var att kontrollera att strömmen fungerade och att kontrollera att knappsatsen fungerade som den skulle. De två lysdioderna är fortfarande en del av hårdvaran, men de är inte med i kopplingsschemat då de inte fyller någon funktion till Whac_a_Mole. Kod skrevs även för att testa att displayen fungerar. När samtliga komponenter var på plats ritades kopplingsschemat i Adobe Illustrator, och användes sedan för att underlätta programmeringen av spelet.

5.3 Programmering

Mjukvaran till Whac_a_Mole har utvecklats i Atmel Studio 7.0 och all kod har skrivits i filen main.c, se *bilaga 2*. Koden som skrivits har delats upp i ett flertal metoder för att möjliggöra metदानrop på flera ställen och för att hålla mainmetoden kort och koncis. Kodens grundstruktur är en switch-sats med tre states: *IDLE*, *GAME_ON* och *TOP_LIST*, som sedan anropar de olika metoderna.

När spelet påbörjas och mainmetoden körs anropas metoden *welcome()*, som välkomnar användarna till spelet och ber dem starta spelet genom att klicka på ON-knappen på knappsatsen. Därefter räknar spelet ner och användarna kan göra sig redo för start.

Under spelets gång lyser lamporna totalt 15 gånger. Det antalet valdes för att en spelare ska vinna i varje omgång samt för att göra spelet lagom långt. Vilken lampa som lyser vid varje tillfälle väljs slumpmässigt och det gör även tiden mellan att lamporna lyser. Tiden mellan att lamporna lyser slumpas med hjälp av *TIMER0* och det valda intervallet är mellan en till fem sekunder. När en spelare klickar på den knapp som motsvarar den tända lampan, slocknar lampan. Därefter jämför programmet de två spelarnas knapptryck, och poäng ges till den som klickat snabbast.

Efter att omgången avslutats presenteras vinnaren, som antingen utgörs av *PLAYER ONE* eller *PLAYER TWO*, på displayen. Därefter presenteras respektive spelares poäng.

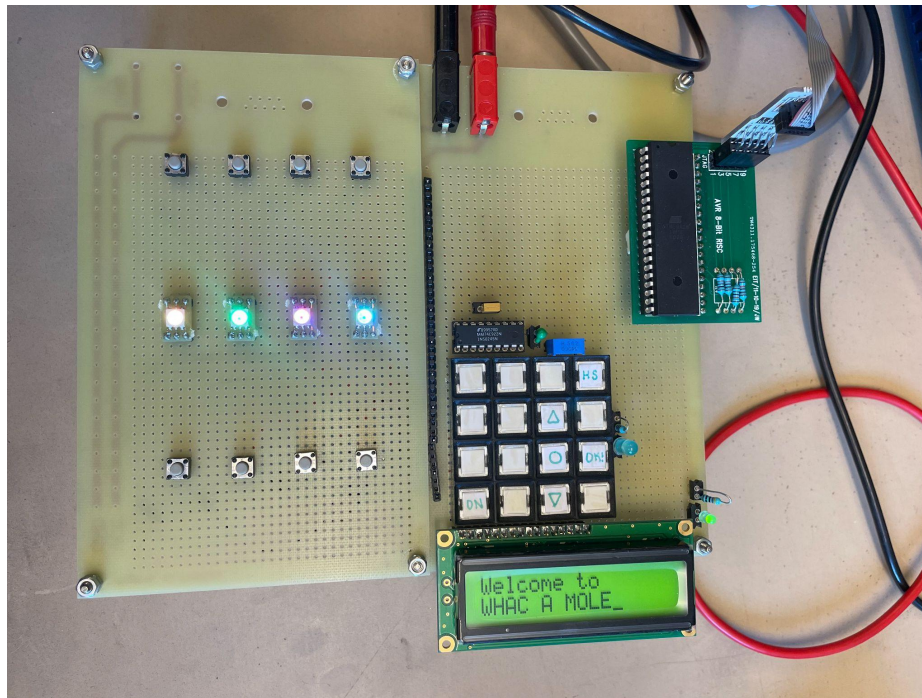
Vinnarens resultat jämförs därefter med det poängrekord som finns lagrat i EEPROM. Rekordet lagras i processorns EEPROM, då det minnet inte nollställs vid reset. EEPROM valdes också framför FLASH-minnet då förändringar i EEPROM kan ske fler gånger innan det slutar uppdateras. Är vinnarens resultat högre än det tidigare rekordet får vinnaren möjlighet att skriva in sitt namn med hjälp av knappsatsen. Slutligen uppdateras rekordet och vinnarens namn samt poäng sparas i EEPROM. På displayen visas därefter det nya rekordet.

När spelet är slut välkomnas användarna till spelet igen, och de får möjligheten att spela en till omgång genom att trycka på ON-knappen. När meddelandet "Start game with the ON-button!" visas, går det även att klicka på HS för att visa det nuvarande rekordet på displayen.

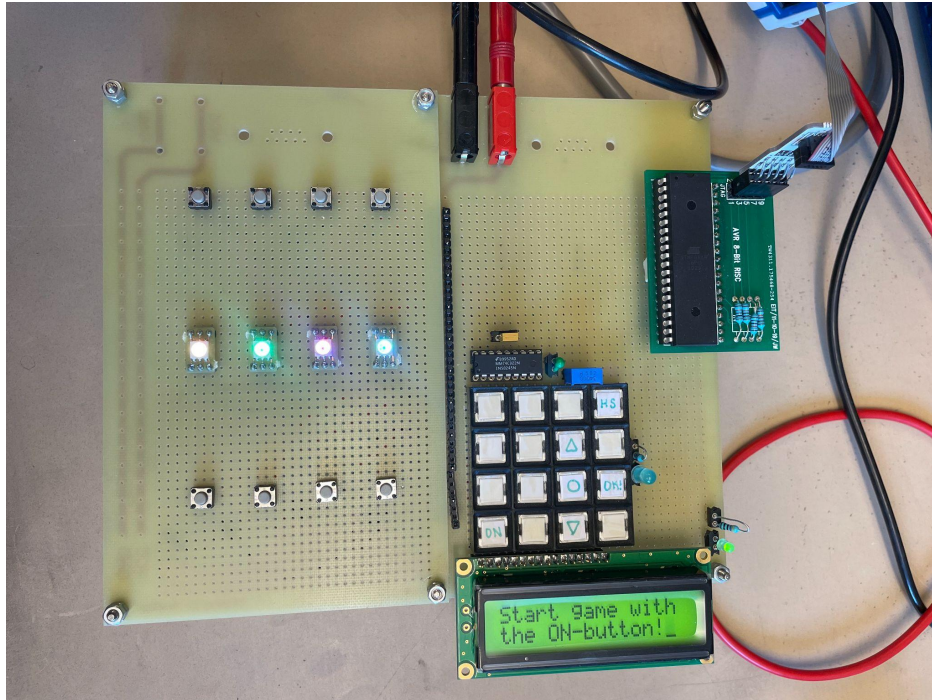
6. Resultat

Nedan presenteras bilder på den slutgiltiga hårdvaran. Bilderna är tagna under alla spelets tillstånd.

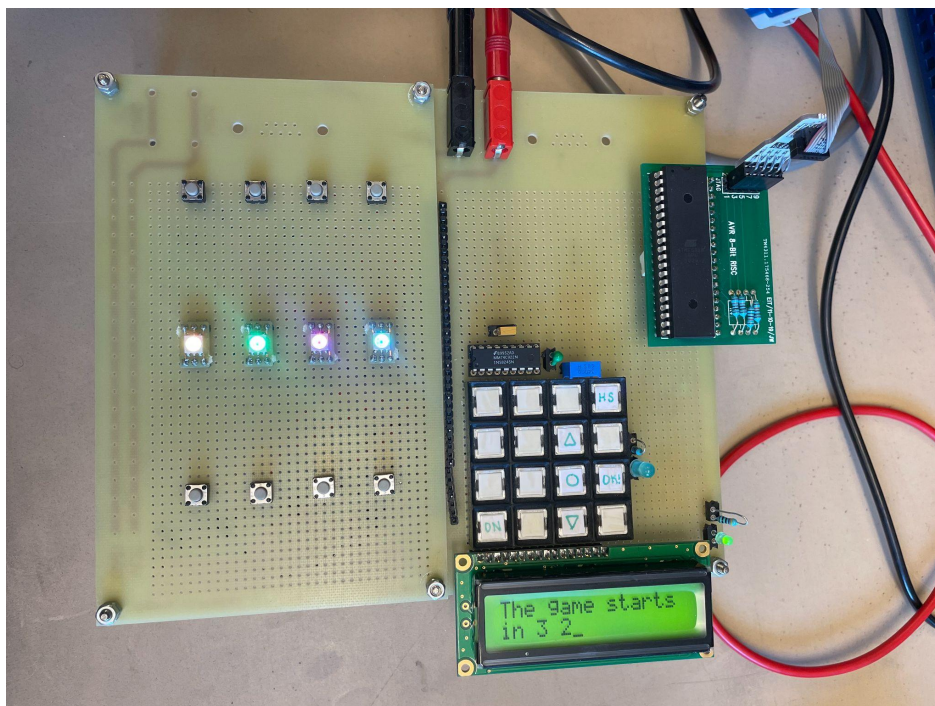
Whac_a_Mole fungerar som önskat och uppfyller samtliga krav som presenteras i *Avsnitt 2. Kravspecifikation*.



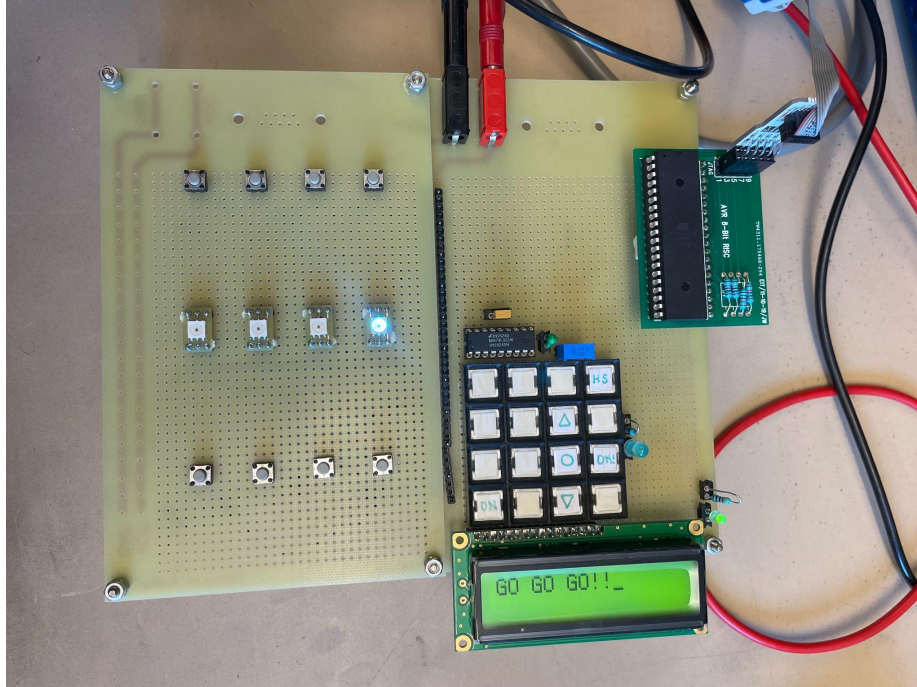
Figur 2. Whac_a_Mole innan ett spel initieras.



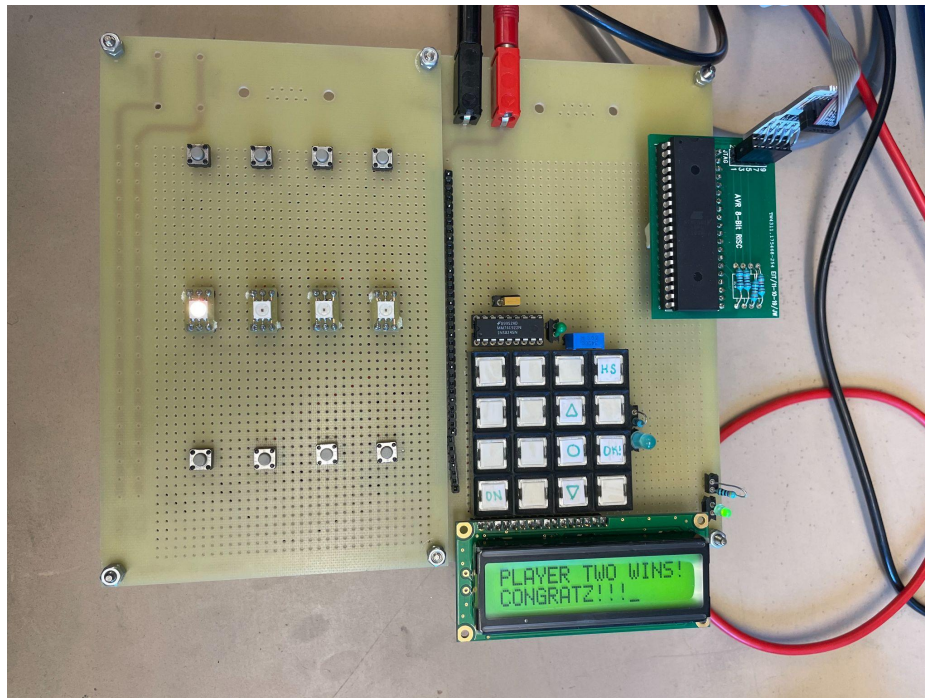
Figur 3. Vänteläget som visas inför att en omgång drar igång. Spelet stannar här tills någon av användarna trycker på ON.



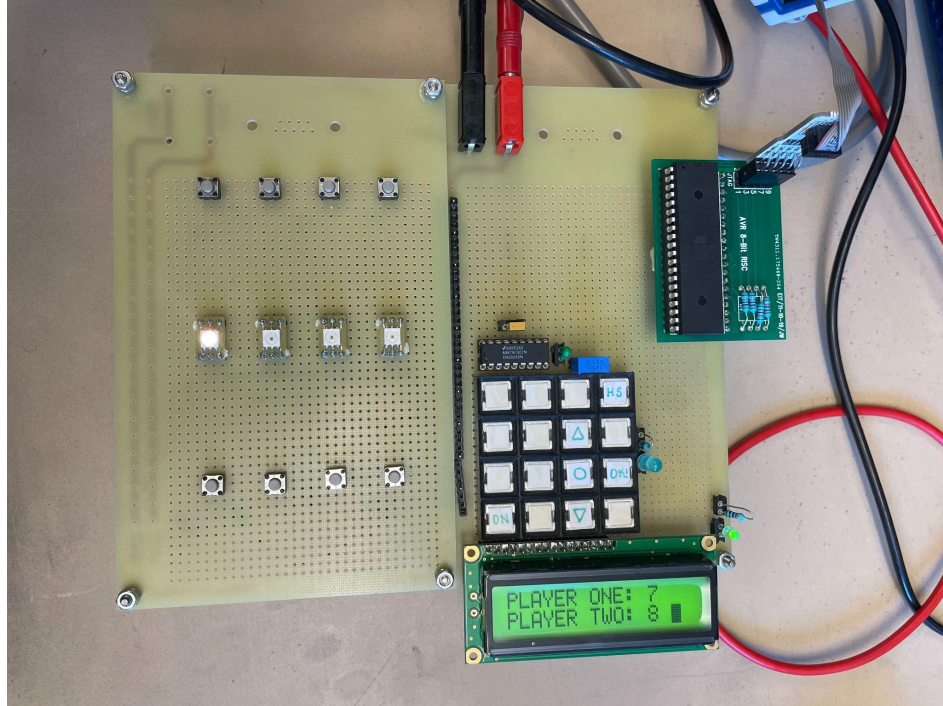
Figur 4. Nedräkningen inför spelets start drar igång när en användare tryckt på ON.



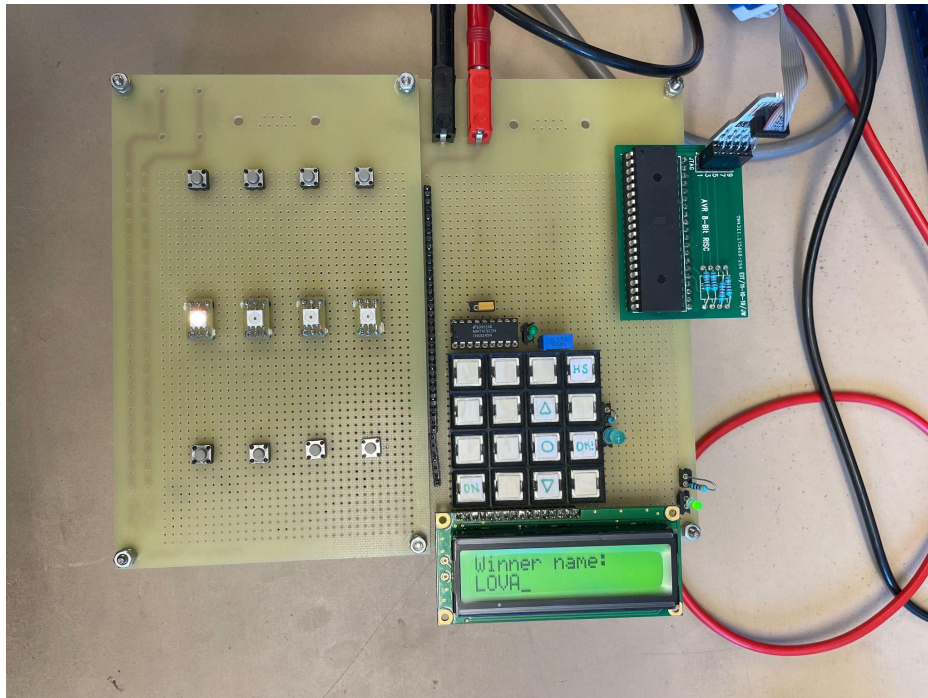
Figur 5. Spelet är igång. En lampa lyser åt gången.



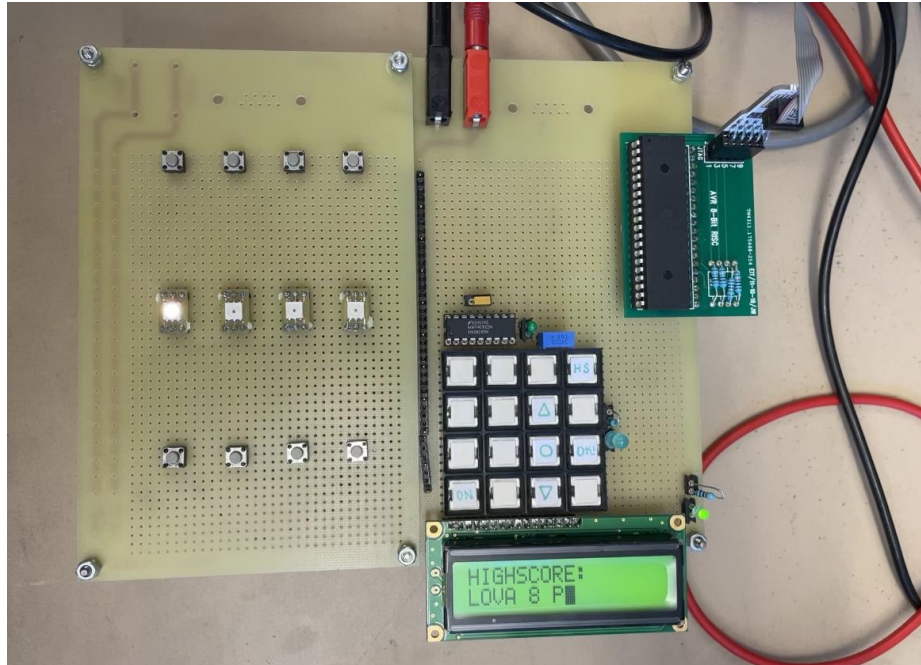
Figur 6. Meddelandet som visas när omgången är slut. I det här fallet har PLAYER TWO vunnit.



Figur 7. Omgångens resultat.



Figur 8. Om vinnaren har slagit rekord får den möjlighet att skriva in sitt namn.



Figur 9. Meddelandet som visas efter att vinnaren skrivit in sitt namn. Detta meddelande visas också vid klick på HS.

7. Diskussion

Projektet har i sin helhet varit både roligt och lärorikt. Däremot har det bitvis varit svårt att ta till sig information, då samtliga gruppmedlemmar har väldigt begränsade tidigare erfarenhet av såväl programmering i C som ellära och assembler. Under projektets gång blev det dock enklare att förstå sig på datablader och komponenter till hårdvaran kunde till slut kopplas ihop utan större svårigheter. För att färdigställa koden behövde vi en hel del hjälp från vår handledare, men även här upplevde vi att mer kod kunde implementeras självständigt eller med hjälp av Google när vi närmade oss projektets slut. Mycket av den kod som skrivits följer den struktur som används i Java, då samtliga gruppmedlemmar har tidigare kunskaper inom det språket.

Under projektets inledande fas fick vi vid ett flertal tillfällen koppla om en del trådar. Det grundas i att vi dragit förhastade slutsatser och i efterhand insett att vi exempelvis inte haft tillräckligt många lediga pins kvar. För att undvika detta i framtiden hade vi kunnat göra kopplingsschemat innan vi påbörjar uppbyggnaden av hårdvaran.

Sammanfattningsvis är vi mycket nöjda med projektet och anser att vi åstadkommit ett väldigt roligt spel, som även följer den kravspecifikation som skapades initialt.

8. Referenser

[1] https://www.eit.lth.se/fileadmin/eit/courses/datablad/Processors/ATmega16_sum.pdf Hämtad 2022-05-17.

9. Bilaga

9.1 Bilaga 1 – Kopplingsschema

■ Kopplingsschema Whac_a_Mole.pdf

9.2 Bilaga 2 – Kod

[Kod Whac a Mole.c](#)