



# LUNDS UNIVERSITET

## Dino Deez

---

*Utveckling av spel i C med ATmega1284 mikroprocessor*

**Rapport Grupp 10**

**Hugo Eklund, Semjons Sapkins, Anh Tri Pham,**

**Ivan Bogosavljevic, Hannibal Lind**

**2023-05-15**

# Sammanfattning

Vi har gjort spelet “Dino deez”. Spelet är gjort i genren 2D-runner och programmeras i språket C. Spelet programmeras på mikroprocessorn ATmega1284 med hjälp av Atmel Studio 7. I vår rapport beskriver vi vår utvecklingsprocess, inklusive de utmaningar vi stött på. Vi beskriver hur vi hanterar dessa hinder och hur vi har löst de tekniska problem som uppstått. Rapporten innehåller detaljer om koden vi skrev och hur vi arbetade i gruppen. Slutligen ger vi en sammanfattning av vår upplevelse av att utveckla “Dino deez”, vårt arbete och vårt resultat.

## Nyckelord

- C-Programmering
- Deez
- ATmega1284
- Kretsschema
- Röststyrd
- Spel

## Abstract

We in group 10 have developed the game “Dino Deez”. The game is a 2D-runner and is programmed in the language C. It is programmed on the ATmega1284 microprocessor using Atmel Studio 7. In our report we describe our development process, including the challenges we faced. We explain how we handled these obstacles and how we solved the technical problems that arose. The report contains details about the code we wrote and how we worked as a group. Finally we provide a summary of our experience developing Dino Deez, our work and our results.

# Keywords

- C-Programming
- Deez
- ATmega1284
- Circuit Diagram
- Voice Controlled
- Game

# Förord

Genom vårt projektarbete har vi som grupp utvecklat vår kompetens och utmanat oss själva inom nya och befintliga arbetsområden. Vi är nöjda med det arbete vi har utfört och upplevde att vår samarbetsprocess fungerade bra. Vi har lärt oss att kommunicera effektivt och att arbeta tillsammans som en grupp, vilket har gjort att vi uppnådde majoriteten av våra mål. Sammanfattningsvis var projektet en utmaning som hjälpte oss att växa både som personer och professionellt.

# Innehållsförteckning

<b>Sammanfattning</b>	<b>2</b>
<b>Nyckelord</b>	<b>2</b>
<b>Abstract</b>	<b>2</b>
<b>Keywords</b>	<b>3</b>
<b>Förord</b>	<b>3</b>
<b>Innehållsförteckning</b>	<b>4</b>
<b>1. Inledning</b>	<b>5</b>
1.1 Bakgrund	5
1.2 Syfte	5
1.3 Kravspecifikation	5
<b>2. Scope</b>	<b>6</b>
2.1 Hårdvara	6
2.2 Mjukvara	6
2.3 Kretsschema	6
<b>3. Teknisk bakgrund</b>	<b>7</b>
2.1 ATmega1284	7
2.2 GDM1286	7
2.3 Kondensatormikrofon	7
2.4 Trimmer Resistor	7
2.5 J-Tag	8
2.6 Mjukvara	8
<b>4. Metod</b>	<b>8</b>
<b>5. Resultat</b>	<b>10</b>
<b>6. Diskussion</b>	<b>12</b>
<b>7. Källförteckning</b>	<b>12</b>

# 1. Inledning

## 1.1 Bakgrund

Dino Deez är ett spel som tar inspiration från ett webbläsarspel som heter "Chrome Dino". Chrome Dino är ett spel där en dinosaurie ska undvika olika objekt genom att hoppa över dem med mellanslagstangenten. Dino Deez bygger på detta koncept genom att återskapa ett liknande spel som är röststyrd, där karaktären styrs av ljud från spelaren som sedan plockas upp av en mikrofon.

## 1.2 Syfte

De huvudsakliga kraven med projektet är att alla mål möts till en godtycklig grad, detta innebär en funktionell krets, hemsida och slutligen fungerande mjukvara för kretsen. Utöver detta krävs det även att arbetet lämnas in i tid inom ramarna för projektets deadline samt att det godkänns av handledaren. Syftet med just vårt projekt var att skapa en fungerande prototyp av ett 2D-plattformsspel med hjälp av C kod samt diverse komponenter. Spelet skulle köras med hjälp av en ATmega1284 mikroprocessor (Atmel Corp. 2016) som sedan visar spelet på en LCD display (GDM 12864) (Xiaomen Ocular Optics). Karaktären ska hoppa över staket med hjälp av en mikrofon som tar in ljudsignaler och - med hjälp av kod i vårt program - får karaktären att hoppa. Uppritningen av ett komplett kretsschema samt programmeringen av en hemsida för uppvisning av projektet är även en del av projektet men för enkelhetens skull fokuserar denna rapport huvudsakligen på spelet & hårdvaran.

## 1.3 Kravspecifikation

- Mikroprocessorn ATmega1284 ska användas.
- En display ska användas för att visualisera spelet.
- Spelet är över när en kollision sker.
- En mikrofon används för att styra karaktären.
- Volymen som mikrofonen tar in påverkar hur högt spelaren hoppar

## 2. Scope

### 2.1 Hårdvara

Projektet ska - för samtliga elever - följa samma grund gällande de komponenter som tilldelas, detta inkluderar kopplingsplattor samt en ATmega1284 mikroprocessor (Atmel Corp. 2016). Utöver det kommer vi använda oss av följande komponenter: 1x GDM12864 LCD display (Xiaomen Ocular Optics) och 1x kondensatormikrofon. Slutligen kommer en J-TAG (Atmel Corp. 2016) att användas för att programmera mikroprocessorn med program skrivna i C, mer om detta nedan.

### 2.2 Mjukvara

För programmering av mikroprocessorn har vi valt att arbeta i verktyget "Atmel Studio 7" och programmet kommer att helt skrivas i programmeringsspråket "C". Målet är att med hjälp av signaler från mikrofonen kontrollera pixlar på LCD skärmen. Vi har i andra hand som mål att - till en realistisk grad - återskapa det ovannämnda "Chome Dino", däremot anser vi att detta är väldigt ambitiöst. Därmed tillåter vi oss själva en generös mängd utrymme för nedskalning av programmet vid behov utan att vara alltför nitiska till planeringen.

Enligt projektets kravspecifikation kommer vi även att programmera en webbplats med hjälp av Javascript, HTML och CSS. Hemsidan kommer att ha med länkar till kretsschemat, den här rapporten samt källkoden till vårt projekt och kommer därmed att agera som en knutpunkt för hela arbetet.

### 2.3 Kretsschema

Uppritningen av ett kretsschema är ytterligare ett grundläggande krav för projektarbetet då det tjänar många fördelar både för läsaren samt vid konstruktionen av kretsen. Kretsschemat agerar som en mall för hur kretsen ska kopplas upp för samtliga komponenter och förbereder därmed läsaren för hur hårdvaran kommer att se ut. Schemat kommer att ritas upp med hjälp av CAD programmet "Eagle".

## 3. Teknisk bakgrund

### 2.1 ATmega1284

En ATmega1284 mikroprocessor (Atmel Corp. 2016) är knutpunkten mellan hårdvaran och mjukvaran i projektet. ATmegan bearbetar datan och logiken som krävs för att driva spelet som att till exempel rita ut pixlar på displayen. Processorn består av 40 pins (se kretsschema) som kan kopplas för att aktivera och kontrollera olika funktioner. D-porten (pin D0 till D5) används för att konfigurera displayen och möjliggöra att data kan skickas fram och tillbaka enligt displayens timing diagram. B-porten (B0-B7) används för att skicka information till LCD:n om var pixlarna ska dyka upp på skärmen. A-porten används för att koppla mikrofonen, så att analoga signaler kan bearbetas till digitala signaler med hjälp av ATmegans 'analog till digital omvandling' (ADC).

### 2.2 GDM1286

LCD:n är en 128x64 LCD grafisk display som kan programmeras för att visa upp enfärgade pixlar på skärmen. Displayen består av 20 pins som används för att exempelvis mata in en drivspänning, variera kontrasten eller läsa in data från ATmegan för att visa pixlar. Skärmen konfigureras med hjälp av en mikroprocessor för att hantera inmatning och utmatning av data enligt displayens specifika timing-diagram. Detta görs genom att bitvis ändra på flaggorna som styr specifika funktioner såsom "chip select" och "read/write" flaggorna. Skärmen är uppdelad i två 64x64 skärmdelar som kan programmeras separat.

### 2.3 Kondensatormikrofon

En kondensatormikrofon är en liten mikrofon som består av två plattor som vid intagning av ljud rör varandra och genererar spänningssignaler. Dessa signaler kan då bearbetas och användas i olika funktioner då man programmerar.

### 2.4 Potentiometer

En potentiometer är en justerbar resistor som kan producera olika motstånd. Resistorerna är skruvbara och kan generera motstånd mellan upp till 10 k  $\Omega$ . Komponenten kan användas för att

justera ljuskontrasten på LCD-displayen eller för att justera att den korrekta mängden spänning matas ut från mikrofonen.

## 2.5 J-Tag

J-tag (Atmel Corp. 2016) är en debugger som agerar som en förbindelse mellan datorn och ATmega. Den består av en port som kopplas till ATmega och används för att programmera processorn.

## 4. Metod

Kopplingsschemat konstruerades enligt databladen för både LCD-displayen och mikroprocessorn. D0-D5 stiftet virades till pin 12-17 på displayen. D-porten ansvarar för att konfigurera displayen och möjliggöra kommunikation mellan displayen och mikroprocessorn. Där specifika flaggor sätts till antingen en nolla eller en etta, enligt timing diagrammet för displayen, för att skriva till eller läsa av data från displayen.

B porten virades från port 4 till 11 på displayen för att kunna skicka 8 bitars pixel data till displayen. Utöver detta lödades två stift till VCC samt mark på kopplingsplattan. AVCC, VCC, VDD och A på processorn och displayen kopplades till en +5V spänningskälla för att tillföra en drivspänning till varje komponent. Alla mark på processorn och K- och VSS stiftet kopplades till mark.

Mikrofonen krävde en RC-krets för att fungera. RC-kretsen består av en resistor (potentiometer) och en kondensator för att fungera som ett filter för frekvenserna som producerades av mikrofonen då ljudinputet matades in. Ett oscilloskop användes för att studera spänningen som producerades av mikrofonen och en kondensator på 10 nanofarad valdes. Mikrofonen producerade maxvärden på ungefär 2V som då kunde matas in till A0 stiftet på processorn. Därefter bearbetas spänningen med hjälp av processorns inbyggda 'analog till digital' (ADC) omvandlare.

Programmering av mikroprocessorn påbörjades med att initiera AD omvandlaren på mikroprocessorn enligt databladen. En prescaler faktor på 128 valdes, samt en referensspänning på 2V då inputen var som högst 2V. Fyra ADC omvandlingar samlades från mikrofonen i ett register och ett medelvärde räknades ut för att då använda som en referensvariabel i andra metoder, till exempel så ska karaktären hoppa då mikrofonen får en tillräckligt stark insignal.



Displayen initierades enligt LCD databladen (Xiaomen Ocular Optics). Specifika flaggor aktiverades genom att skicka data bitvis via D-stiften för att kunna initiera read och write funktionen till displayen. Metoder skapades i C för varje flagga såsom "chipSelectLow" (en nolla skickas till chipSelect biten) eller "chipSelectHigh" (en etta). Som då aktiverades i koden för att använda displayen.

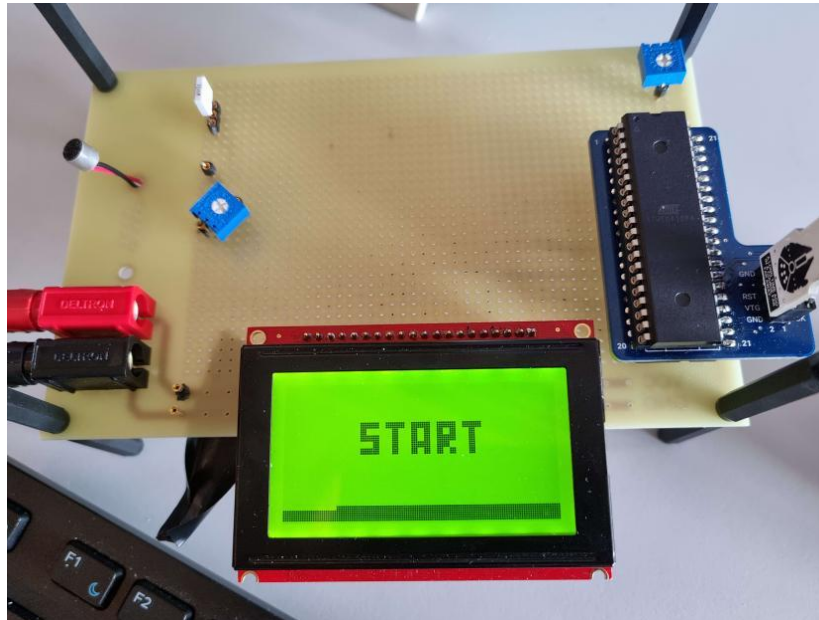
Metoden "draw" och "setPixel" utvecklades därefter för att kunna rita upp pixlar på LCD-displayen. D-port metoderna användes för att initiera skrivning till displayen. I "draw"-metoden delades skärmen upp i åtta celler och x koordinater. Om en x-koordinat skickades in som är över 64 hamnade den på den delen av skärmen som är längst till höger, och tvärtom skickas data till vänstra delen. Datan samlas och skickas då vidare via B-porten. 2 draw metoder utvecklades, en som har en funktion där varje objekt som ritas upp på skärmen raderas och en där objekt är statiska.

Utöver denna metod utvecklades "SetPixel"-metoden, som tar in en x- och y-koordinat, samt en "scale" inparameter som kontrollerar dimensionerna på objektet som skapas på skärmen.

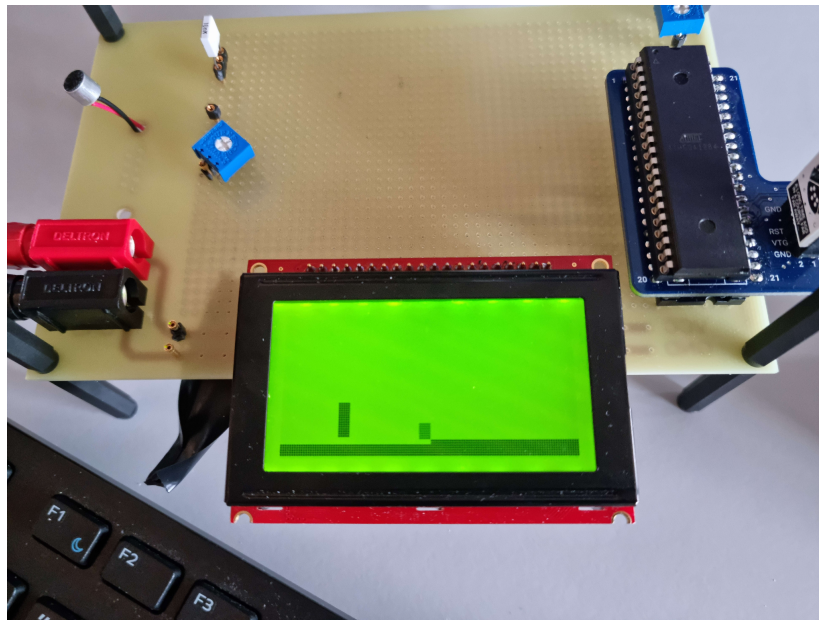
Själva spelinnehållet fanns med i "fence"-metoden som innehöll en for-loop som ritade upp ett objekt som rörde sig från höger till vänster. Objektet blev hindret för karaktären att hoppa över. Inuti "fence"-metoden fanns en if-sats som jämförde mikrofoninput med ett referensvärde på ungefär 600-800. Om värdet från mikrofonen var högre än referensvärdet skulle karaktären ritas upp i mitten av skärmen, annars ritades karaktären upp vid botten av skärmen. Detta användes för att simulera hoppet. Utöver detta programmerades "terrain"-metoden som skrev ut en liten remsa av mark på botten av skärmen. Slutligen programmerades en "START" och "GAME OVER" skärmbild, som skulle dyka upp på skärmen när spelet startas eller om spelet förloras (då karaktärens x- och y-koordinat överlappar med staketet som rör på sig).

## 5. Resultat

Resultatet är ett relativt väl fungerande "Dino spel" med en Start meny (se bilaga 1) som startar spelet genom en mikrofoninput. En terräng ritas upp och karaktären är placerad i en statisk position för att vara beredd att hoppa över ett motsvarande hinder, i detta fall är det en pixel som rör sig mot karaktären (se bilaga 2). Om karaktären misslyckas att hoppa över hindret med hjälp av mikrofonen - d.v.s. om både karaktären och hindret kolliderar med varandra - visas "Game Over" över skärmen i ett par sekunder innan start-menyn visas igen (se bilaga 3).



*Bilaga 1: En bild på start-menyn*



*Bilaga 2: Den rektangulära figuren åt vänster motsvarar vår karaktär.  
Den lilla kvadraten åt vänster är hindret och är på väg mot figuren*



*Bilaga 3: Game Over menyn*

## 6. Diskussion

Vi bollade lite idéer i början men det gick väldigt snabbt till att hitta en idé som fungerade för alla i gruppen. Gruppen var enig om att det skulle göras något typ av spel och det som vi kom fram till var att göra en kopia av Chromedino spelet som spelas på Chrome.

Arbetet från kretsschema till att löda och vira allt till breadboard har fungerat utan problem förutom att vi i början virade kablarna till displayen spegelvänt. Det löstes ganska enkelt genom att rotera displayen 180 grader. Men efter det lilla problemet stämde kretsschemat perfekt till verkligheten.

När det kom till själva programmeringen av både LCD och spelet fick vi vårt första problem i början. Det som mest strulade var att få själva LCD att fungera. Vi visste inte riktigt hur vi skulle skicka pixlarna till displayen då man måste koda varje pixel individuellt. Det som tog längst tid på projektet var att förstå LCD och skriva fungerade kod till spelet. Men när vi väl fick det gjort tog det inte särskilt lång tid för oss att fixa allt vi ursprungligen hade tänkt att ha med i spelet.

Något som var med på kravspecifikationen i början var att kunna styra hur högt karaktären kan hoppa beroende på hur högt du ropar in i mikrofonen. Detta märkte vi ganska snabbt var en svår implementerad idé eftersom att när vi testade mikrofonen med ett oscilloskop fick vi inte några utslag oavsett hur nära eller högt man skrek. Det som fungerade bäst och gav konsekventa utslag var att blåsa rakt in i mikrofonen vilket vi tyckte var okej som ersättning till vår originala idé.

## 7. Källförteckning

1. *Atmel Corporation, "ATmega1284 - Datasheet", 42718C, 10/2016*
2. *Atmel Corporation, "The Atmel-ICE J-Tag (User Guide)", 42330C, 10/2016*
3. *XIAMEN OCULAR OPTICS CO.,LTD, "User's Guide GDM12864HLCM (Liquid Crystal Display)"*