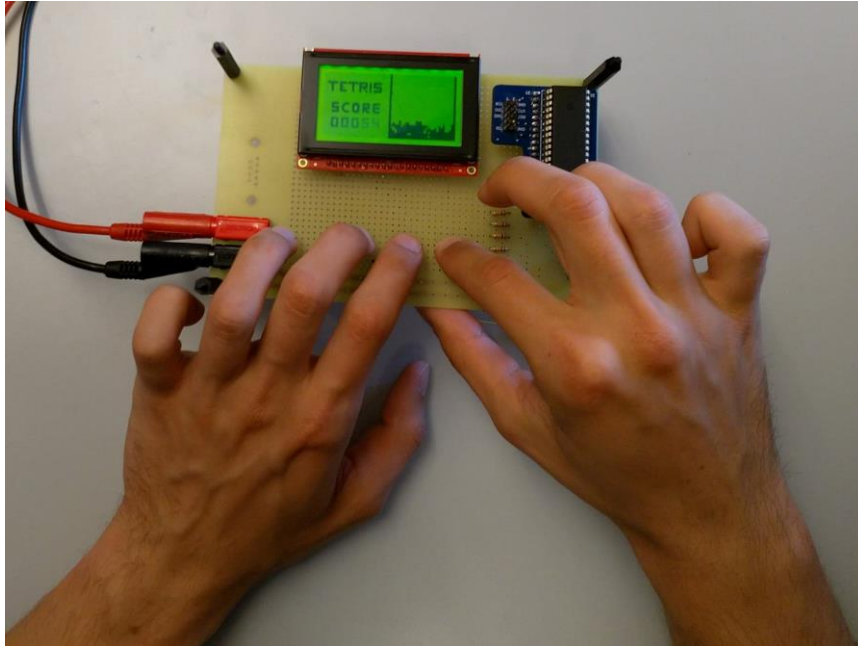


TETRIS



LTH, Campus Helsingborg
EITA15 Digitala System

Handledare: Bertil Lindvall
Författare: Isak Shamun, Viktor Kulle, Mark Slipac och Dennis Järnåsen

Datum: 2019-05-09

Abstract

This report concerns the realization of a Tetris-based game on the 8-bit ATmega1284 microcontroller as a part of the course Digital Systems. It encompasses the entire working process, from setting the constraints of the prototype, to ideas for potential upgrades to it.

The program code is in C and the interfacing hardware is composed of an LCD-display and four buttons.

Sammanfattning

Denna rapporten behandlar realiseringen av ett Tetris-baserat spel på mikrokontrollern ATmega1284 som projekt i kursen Digitala System. Rapporten omfattar hela arbetsgången, från kravspecifikationen till utvecklingsmöjligheter för prototypen. All programkod är skriven i C och användargränssnittet består av en LCD-display och fyra tryckknappar.

Nyckelord

AVR, ATmega1284, Projekt, C, Digitala system, Spel, Tetris.

Innehållsförteckning

1. Inledning.....	1
1.1 Kravspecifikation.....	1
2. Komponenter.....	2
3. Metod.....	3
3.1. Planering.....	3
3.2. Konstruktion	3
4. Resultat	4
5. Diskussion.....	4
6. Källförteckning.....	6

1. Inledning

Som projekt i kursen Digitala System bestämde vi oss för att realisera spelet Tetris på en ATmega1284 i syfte att lära oss mer om AVR-programmering, hur man kopplar diverse komponenter samt implementera algoritmerna som krävs för spelets funktioner.

Syftet med projekt-delen i kursen var att tillämpa kunskaperna vi har erhållit från laborationerna för ATmega1284 för att bygga vår egen prototyp från grunden.

1.1 Kravspecifikation

1. Det ska finnas en skärm till konsolen
2. Konsolen ska ha en höger-, vänster, ner- och rotationsknapp.
 - a. Vid nedtryckning av höger-/vänsterknappen ska blocket flyttas ett steg åt det håll så länge det finns plats på spelplanen
 - b. Vid nedtryckning av uppknappen ska blocket vridas ett kvarts varv.
 - c. Vid nedtryckning av nerknappen ska blocket släppas rakt ner.
3. Konsolen ska vid start rita upp en meny med en startfunktion som startar spelet.
4. Konsolen ska kunna rita upp samtliga Tetris-block.
5. Spelet ska avslutas då något block överskrider den översta raden på spelplanen.
6. Displayen ska visa titeln "TETRIS" samt spelplanen under spelets gång.
7. En poängtavla som ska visa poängen under spelets gång.

2. Komponenter

Mikrokontroller

Mikrokontrollern som användes i konstruktionen var en ATmega1284. Atmega1284 är en 8-bitars mikrokontroller med 44 pins varav 32 är I/O portar samt ett flashminne på 128KB.

LCD-skärm

GDM12864 var skärmen som användes till konstruktionen av konsolen. Skärmen har bakgrundsbelysning tillgänglig och har dimensionerna 128x64 pixlar där varje hälft styrs separat.

Knappar

Fyra knappar är anslutna till en pin i processorn som genererar ett avbrott vid nedtryckning. När en knapp inte är nedtryckt så är spänningen till processorn noll Volt då den är kopplad via en resistor till jord.

Varistor

En varierande resistor är ansluten till LCD-skärmen för att reglera arbetsspänningen.

Resistor

4 resistorer på 10K ohm används till knapparna.

JTAG

JTAG används för att kunna felsöka och ladda över programmet till mikrokontrollern.

3. Metod

3.1. Planering

Projektet började med diskussion kring vilket spel som skulle kunna konstrueras samt problemformulering kring de olika förslagen. Efter en röstning enades författarna om att realisera spelet "Tetris". Utifrån detta gjordes en kravspecifikation som sedan diskuterades med handledaren. Planering av framtida möten och labbtillfällen planerades via Facebook då detta ansågs vara enklast. Utrustning och komponenter bestämdes utefter kravspecifikationen och därefter ritades kopplingsschemat i ritprogrammet Autodesk Eagle.

3.2. Konstruktion

Då kopplingsschemat godkändes av handledaren tilldelades gruppen komponenter och utrustning. Komponenterna monterades på mönsterkortet och kopplades samman enligt kopplingsschemat med både virning och lödning för att bilda en prototyp, se bilder 1 och 2 nedan. Då en så kallad "sanity check" hade genomförts med debuggern så säkerställdes det att konstruktionen var korrekt. Kopplingsschema för prototypen finns som Bilaga 1 i slutet av rapporten.

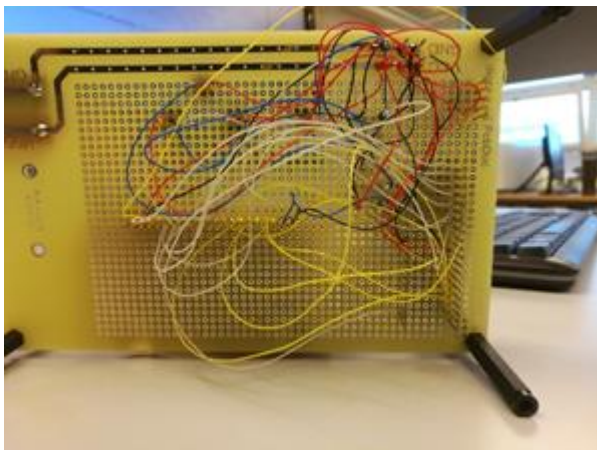


Bild 1: Baksidan av mönsterkortet som visar virningen av alla sladdar.

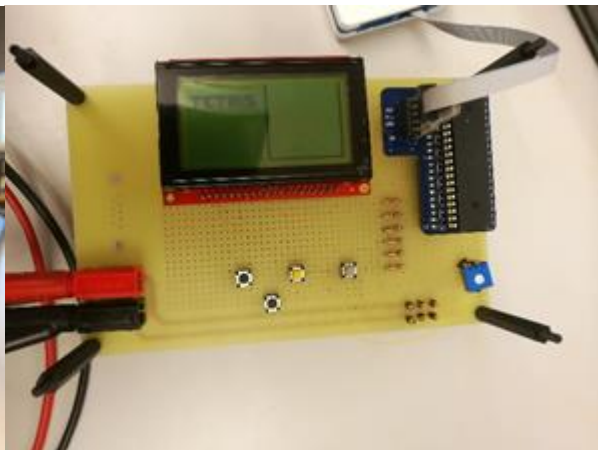


Bild 2: framsidan av mönsterkortet som visar dispositionen av display och knappar, samt lödningen av vissa komponenter.

3.3. Program

Med en fungerande konstruktion påbörjades programmeringen i programmet Atmel Studios. Arbetsprocessen delades upp i 4 steg. Första steget bestod av att kunna sätta igång displayen, andra steget bestod av att rensa information från skärmen och det tredje steget bestod av att tända individuella pixlar på skärmen. Till sist var det dags att implementera spelets funktioner och regler i programmet.

Flödesschema för programmet finns som Bilaga 2 i slutet av rapporten.

4. Resultat

Resultatet har överträffat förväntningar då en poängtavla har lagts till på den vänstra sidan av skärmen som räknar antal rader rensade samt en "Game Over" rubrik som dyker upp när man förlorar. Poängtavlan ökar med ett för varje block placerat och ger extra poäng vid radrensning utefter antal rader som rensas samtidigt.

Se bild 3 nedan för slutprodukten.



Bild 3: Bild av den resulterande prototypen

5. Diskussion

De största motgångarna under projektets gång har varit att skriva fungerande metoder för skrivning till och rensning av skärmen, samt att förstå hur koordinatsystemet för skärmen fungerar. Majoriteten av motgångarna var lösbara genom tillägg av fördröjningar och avbrott i lämpliga ställen i programmet. Den viktigaste lärdomen har varit att lära sig hur man använder debuggern för att få en idé om var exekveringen hakar upp sig eller inte beter sig som man hade tänkt.

Potentiella förbättringar för prototypen hade kunnat vara ytterligare funktioner som finns i de mest förekommande Tetris-spelen idag, till exempel:

- Ett mer avancerat rensningssystem där man särskiljer olika sorters radrensningar genom att meddela detta i spelet.
- Rekordtabell där man sparar den högsta uppnådda poängen.
- Flera nivåer som successivt ökar hastigheten som Tetris-blocken faller
- Högtalare som spelar musik, musikens tempo ska öka med nivåer.
- Ett "battle-mode" där man kan spela mot datorn eller en annan spelare. Om en spelare rensar minst en rad så hamnar det antalet rader på motståndarens spelplan under motståndarens placerade Tetris-block. Dessa så kallade "garbage lines" kan också rensas.

6.Källförteckning

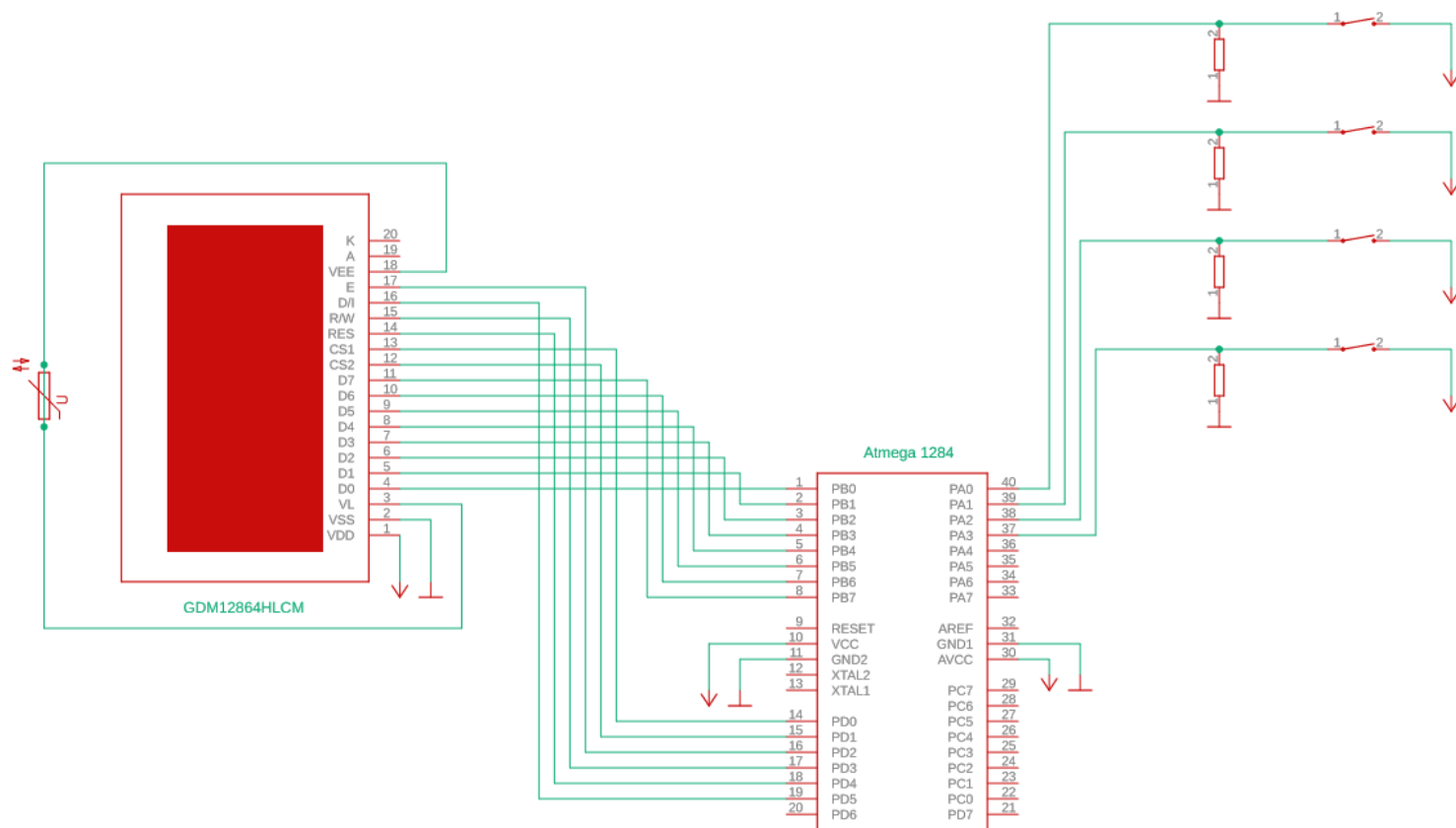
Datablad Display, GDM12864HLCM.

<https://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Display/GDM12864H.pdf>

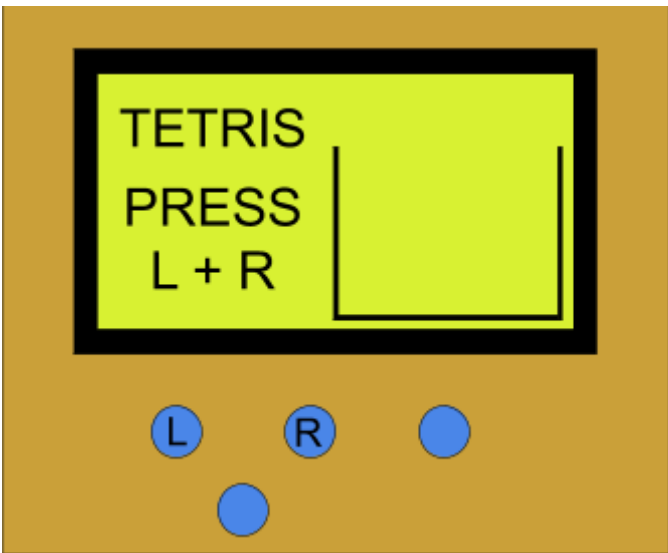
Datablad Processor, ATmega1284.

<https://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega1284.pdf>

7.Bilagor

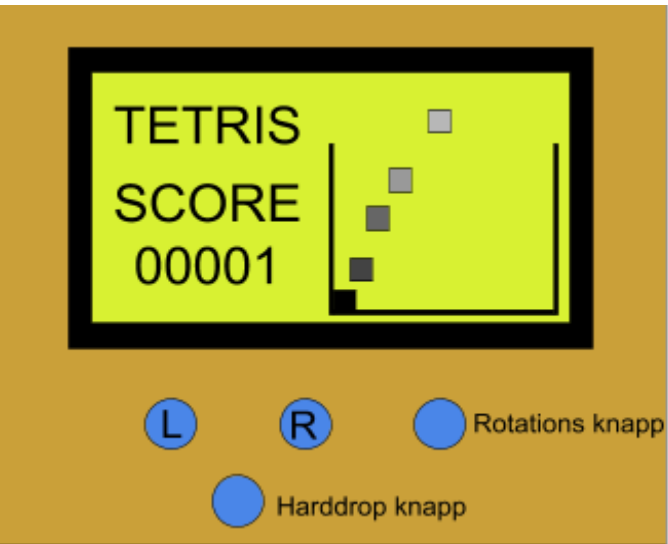


Bilaga 1: Kopplingsschema för prototyp



När strömmen slås på sker följande:

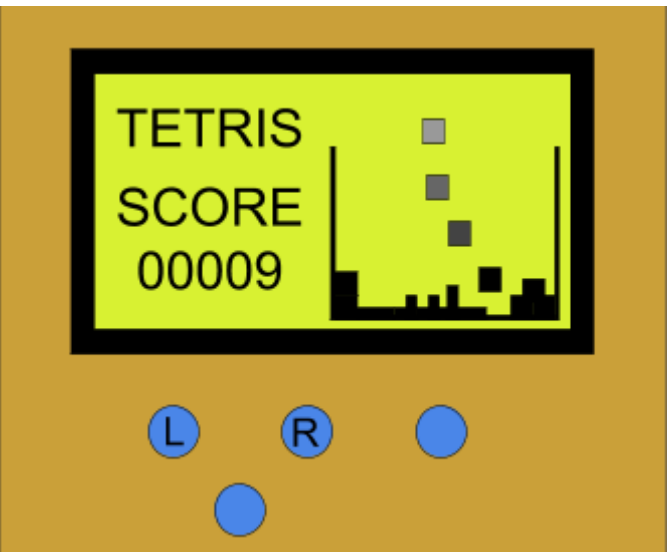
- Antal Tetris-block ställs in till 7
- IO-portar ställs in
- Displayen startar
En metod rensar skärmen på eventuella fel, från höger till vänster
- Spelplanen ritas enligt bilden



Spelaren håller ner L + R för att starta spelet och då ritas poängtavlan ut som har värdet 0000 från början.

Därefter kan spelgången se ut på detta viset:

- Ett slumpmässigt block ritas och faller långsamt ner.
- Användaren kan förflytta blocket med knapparna.
- Trycker man på "hard-drop-knappen" faller blocket rakt ner väldigt fort utan att användaren kan förflytta blocket i sidled.
- Blocket landar och spelaren får ett poäng.



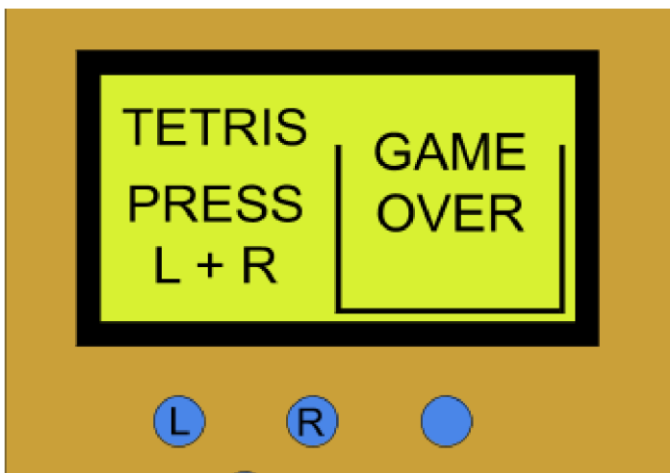
Spelaren har nu fått ner 9 block och således samlat ihop 9 poäng, efter detta blocket kommer raden rensas och spelaren belönas med ytterligare poäng.



Spelaren har inte lyckats rensa några rader och har således byggt för högt. När de staplade blocken överskrider spelplanens höjd, har spelaren förlorat och spelet slutar köra omgången.



Efter förlust ritas "GAME OVER" över spelplanen och poängtavlan står kvar i fem sekunder.



Efter fem sekunder får spelaren möjlighet att starta en ny omgång genom att hålla ner vänster- och högerknapp.