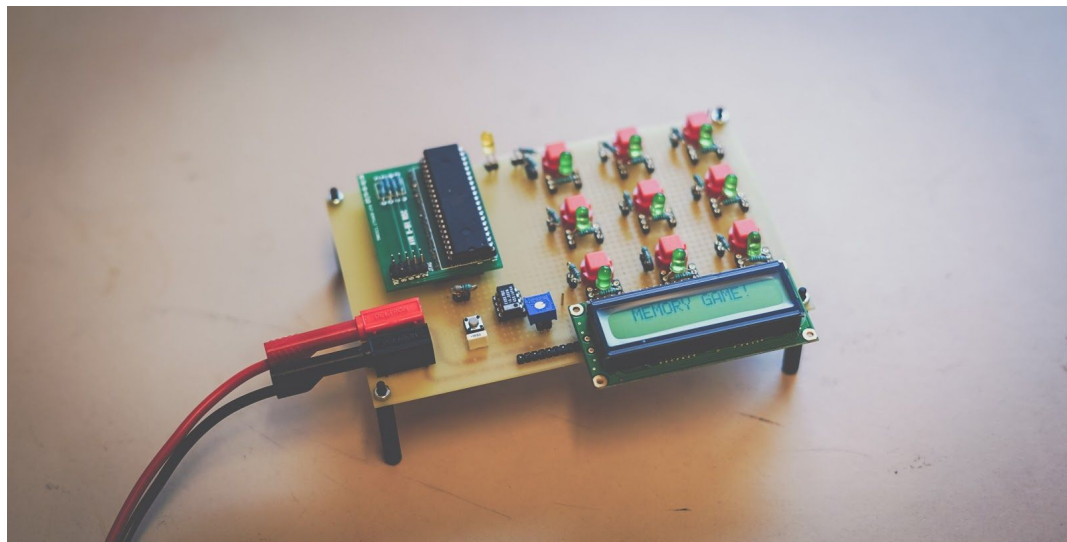


# Projektrapport

## Memory Game



**Program:** Industriell Ekonomi, LTH

**Kurs:** EITF12 Digitala projekt

**Elever:** Alv Romell, Magdalena Karlsson Nycander, Ella Jonsson

**Handledare:** Bertil Lindvall & Christoffer Cederberg

**Datum:** 2019-05-22

# Innehållsförteckning

<b>1. Inledning</b>	<b>3</b>
1.1 Bakgrund	3
1.2 Syfte	3
<b>2. Kravspecifikation</b>	<b>3</b>
<b>3. Hårdvara</b>	<b>4</b>
<b>4. Mjukvara</b>	<b>5</b>
<b>5. Metod</b>	<b>5</b>
5.1 Planering	5
5.2 Konstruktion	6
5.3 Programmering	6
<b>6. Resultat</b>	<b>7</b>
<b>7. Diskussion</b>	<b>11</b>
7.1 Svårigheter	11
<b>8. Referenser</b>	<b>12</b>

# 1. Inledning

## 1.1 Bakgrund

I kursen EITF12 Digitala Projekt ges studenter möjligheten att konstruera en hårdvara parallellt med implementationen av tillhörande mjukvara. Denna rapport behandlar ett sådant projekt där användaren kan testa och utmana sin minnesförmåga i ett spel. Denna konstruktion definieras och utformas utifrån kravbilderna beskriven i avsnitt 2.

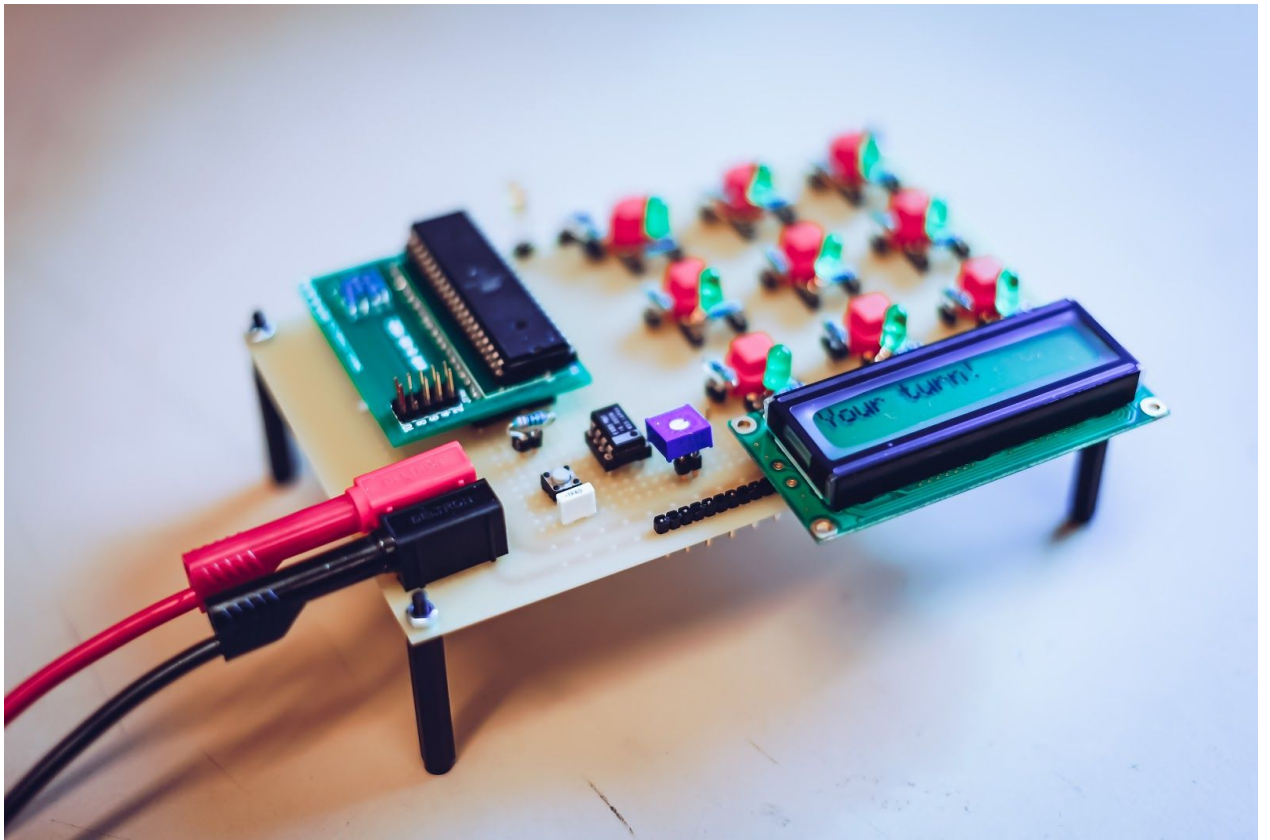
## 1.2 Syfte

Syftet med projektet är få en förståelse för industriellt utvecklingsarbete genom att utveckla en prototyp för vidareutveckling med nödvändig dokumentation.

# 2. Kravspecifikation

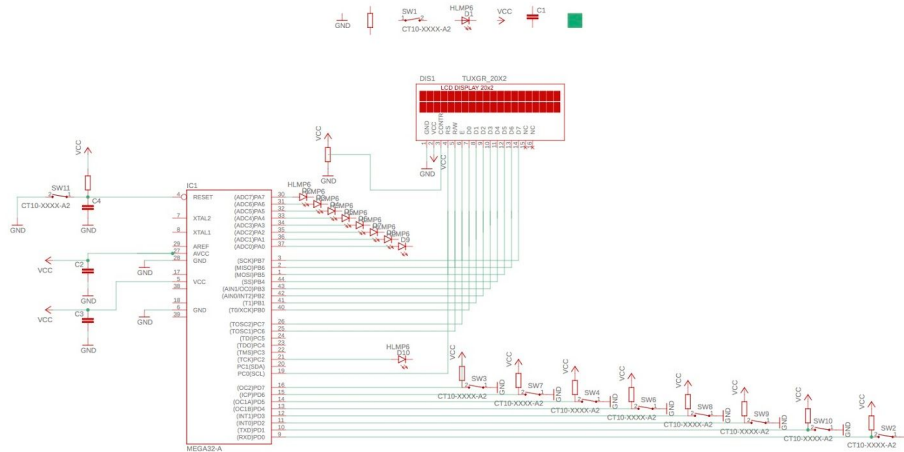
- ❑ Display innan spel visar en startmeny där spelaren får välja att starta spelet.
- ❑ Spelet ska tända lampor i en följd. Spelaren ska sedan ges möjlighet att återskapa denna följd genom att trycka på knappar motsvarande lamporna i rätt ordning. Varje gång spelaren trycker på en knapp ska motsvarande lampa tändas upp som feedback.
- ❑ Om spelaren återskapar följden av lampor korrekt går denne vidare till nästa nivå. Varje nivå ska ha en slinga där en mer lampa ska lysa jämfört med föregående nivå.
- ❑ Klarar spelaren inte att återskapa följden korrekt avbryts spelet och spelarens slutgiltiga poängen visas på skärmen.
- ❑ Spelet ska spara de tre bästa spelarnas poäng på en topplista. Körs spelet och spelaren får lika mycket mycket eller högre än någon på topplistan ska spelaren ges möjlighet att skriva in sitt namn varpå topplistan uppdateras med detta namn och motsvarande poäng. Oavsett om spelaren tagit sig in på topplistan eller ej ska topplistan skrivas ut på skärmen efter att spelaren slutgiltiga poängen skrivits ut.

### 3. Hårdvara



*Bild 1 visar den slutgiltiga hårdvaran.*

- Kopplingsschema



2019-05-21 14:33 f=0.80 /Users/ellajonsson/Desktop/EITF12/Kopplingsschema (1).sch (Sheet: 1/1)

Bild 2 illustrerar det kopplingsschema som hårdvaran konstruerats utefter.

- Processor  
I detta projekt användes en ATmega16 processor, vilket är en 8-bitars mikrokontroller baserad på AVR-förstärkt RISC-design. [1] Processorn programmeras i programspråket C i *AtmelStudio*, vilket överförs via en JTAG.
- Display  
Som display används en bakgrundsbelyst LCD *HD44780 Compatible* med 2 rader á 16 tecken och måtten 80x36mm. [2]
- Lysdioder  
Hårdvaran har en matris av 3x3 lysdioder som var och en är kopplade till processorn genom en resistor. Lysdioderna är RND 135-00038 LED-lampor som är gröna och har en diameter på 5 mm. [3]
- Knappar  
Till varje lysdiod hör en mekanisk knapp som vid nedtryckning ändrar spänningsnivå och på så sätt skickar signaler till processorn. Därutöver finns en reset-knapp som stänger av respektive startar spelet.

- Resistorer  
En 10 k $\Omega$  resistor har kopplats till vardera knapp och lysdiod och konfigurerats som en pull up-resistor.

## 4. Mjukvara

- Autodesk EAGLE  
Framtagning av kopplingschema
- Atmel Studio 7.0  
Programmering av mjukvara och program.

## 5. Metod

### 5.1 Planering

Inledningsvis bestämdes vad vårt projekt skulle bestå av utifrån givna förutsättningar. Flera olika typer av idéer övervägdes, och vi beslutade att utveckla ett spel efter konsultation med handledare. Därefter togs en initial kravspecifikation fram som användes som underlag för att ta fram ett första kopplingsschema. Detta kopplingsschema förbättrades sedan med hjälp av feedback från handledare. Kopplingsschemat gjordes i programmet Eagle.

### 5.2 Konstruktion

När kopplingsschemat var färdigställt påbörjades hårdvarans uppbyggnad. Första steget var att löda fast olika komponenter såsom processor, resistorer och pins till exempelvis ground och VCC. Detta utfördes med hjälp av lödtråd och en magnestat-lödkolv. Därefter användes en virpistol för att koppla processorn till VCC och ground enligt kopplingsschemat på bild 1. Sedan löddes knappar, lysdioder och display fast och kopplades på motsvarande vis.

Den initiala tanken var att använda seriekopplade neopixlar istället för lysdioder eftersom vi inte hade tillräckligt med portar för att ha både 9 knappar och 9 lysdioder inkopplade. Hårdvaran färdigställdes med neopixlar men trots grundlig testning, felsökning med hjälp av handledare samt test med två olika typer av neopixlar så fick vi det inte att fungera. På grund av detta byttes neopixlarna i samråd med handledare tillslut ut mot lysdioder ändå och kopplingsschemat uppdaterades så att varje lysdiod fick en egen port på processorn. Då det inte fanns tillräckligt med portar för att ha både 9 lysdioder och 9 knappar inkopplade valdes det att koppla ur en knapp. Att ha alla lysdioder inkopplade prioriterades över att ha alla knappar inkopplade eftersom en knapp utan motsvarande lysdiod hade varit oanvändbar, medan den nionde lysdioden kan fylla en funktion då att alla lysdioder då kan tändas vid highscore. I slutändan valdes ändå den nionde knappen att behållas av estetiska skäl.

## 5.3 Programmering

Mjukvaran har utvecklats i Atmel Studio och delats in i 6 source-filer med tillhörande headerfiler samt en extra headerfil, `game_matrix.h` som används för att "slumpa" en vektor med lysdioder som ska tändas. Matrisen har 30 rader med 200 element vardera.

Denna uppdelning gjordes för att få en bättre översikt av koden. Spelet konstruerades på ett sådant sätt att när en ny nivå startas så väljs en vektor ur matrisen i `game_matrix.h`-filen, samt ett index i denna vektor, utifrån tidpunkten då nivån startas vilket mäts med hjälp av `TIMER1`.

Denna vektor innehåller siffror mellan 1-8 vilket motsvarar vilken lysdiod som ska tändas varpå denna vektor ger en slinga av lysdioder som tänds med korta mellanrum.

Efter att lysdioderna har tänts ges en uppmaning att det är för spelaren att återskapa denna slinga genom att trycka på motsvarande knappar. Varje knapptryckning jämförs mot den tidigare slumpade lysdiodsvektorn för att avgöra om man tryckt rätt. Har spelaren tryckt rätt får denne feedback i form av att "OK" skrivs ut på displayen. Lyckas spelaren trycka rätt på hela sekvensen skrivs "OK, new level" ut på displayen och en ny nivå initieras. Knapptryckningarna registreras genom att vi mäter av alla knappar vid regelbundna intervall.

Detta görs med `TIMER0`, som är en 8-bitars timer. Var femte overflow (där vår klockfrekvens är satt till 8 MHz och vi använder en prescaler satt till 1024), avgörs om någon knapp är nedtryckt.

Vid en felaktig knapptryckning skrivs "GAME OVER" ut på displayen. Vid denna tidpunkt jämförs spelarens poäng mot topplistan som hämtas ut EEPROM-minnet. Har spelaren tagit sig in på topplistan får denne feedback i form av att "HIGHSCORE" skrivs ut samt att alla lysdioder blinkar. Spelarens poäng sparas sedan i topplistan i EEPROM-minnet. Oavsett om spelaren kommit in på topplistan skrivs topplistan ut, en efter en, på displayen.

Topplistan lagras i EEPROM-minnet för att kunna sparas långvarigt då EEPROM ej nollställs vid en reset.

Spelet har olika tillstånd som avgör vilken del av programmet som ska köras vid respektive tillfälle. Detta gör det möjligt att navigera mellan aktivt spel, topplista och spelslut smidigt beroende på hur användaren agerar.

Uppdelning av kod har gjorts i följande filer

- **`main.c & glob_header.h`**  
Mainfilen använder resterande filer och hanterar de olika tillstånden i spelet. Headerfilen `glob_header` initierar klockfrekvensen.
- **`gameplay.c & gameplay.h`**  
Gameplayfilen startar spelet.
- **`display.c & display.h`**

Sourcefilen display används i main sig av för att skriva ut meddelanden på skärmen.

- **button.c & button.h**  
Sourcefilen button med tillhörande headerfil tillhandahåller kod för att läsa av vilken knapp som blir nedtryckta vid vilken tidpunkt.
- **LED.c & LED.h**  
Sourcefilen LED och tillhörande headerfil hanterar lysdioderna och deras tillstånd.
- **highscore.c & highscore.h**  
Highscorefilen används för att lagra de tre bästa resultaten i EEPROM-minnet för att kunna spara dessa mellan rundor även om man stänger av spelet.
- **game\_matrix.h**  
Headerfilen game\_matrix innehåller vektorer med de slumpmässiga sekvenser som lysdioderna lyser i. Dessa togs fram med hjälp av Matlab där 30 stycken vektorer med längden 200 och siffror mellan 1-8 slumpades.

## 6. Resultat

Nedan presenteras den färdiga prototypen. Den slutgiltiga produkten uppfyller samtliga krav i kravspecifikationen. Nedan visas några utav spelets tillstånd.

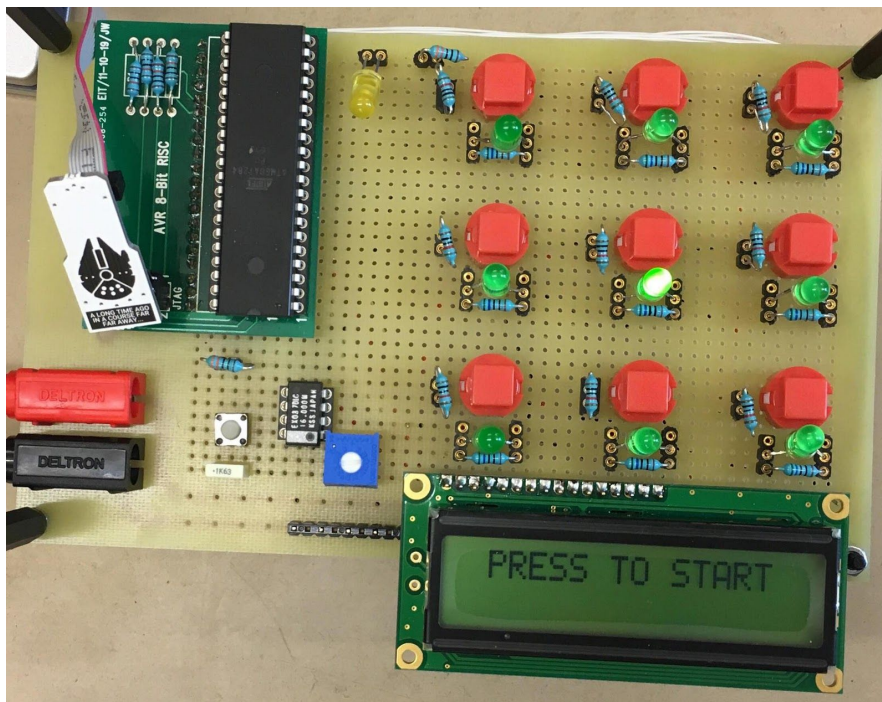


Bild 3 visar tillståndet innan spelet påbörjats.



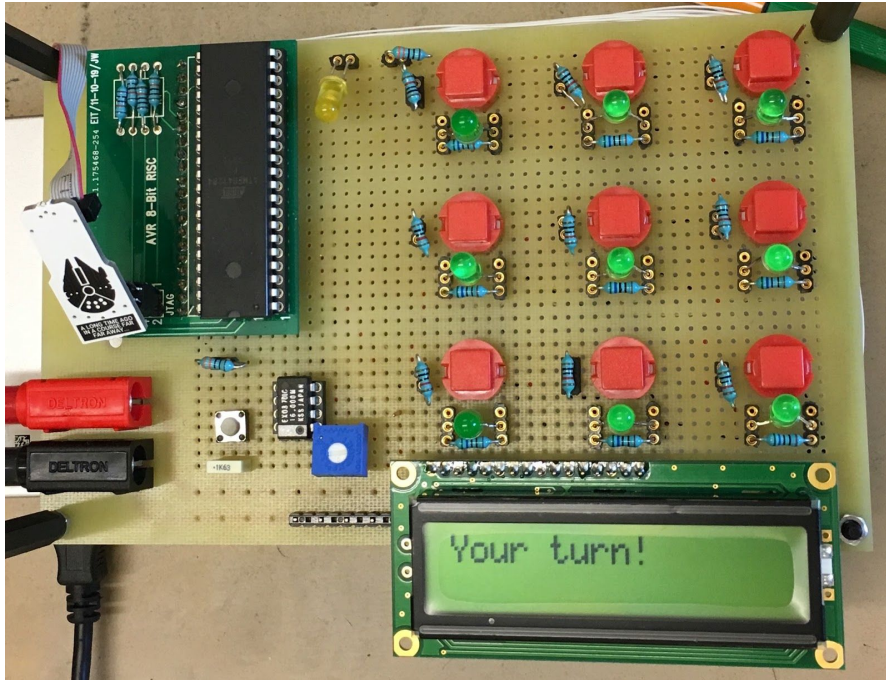


Bild 4 visar när spelet är igång och användaren ska härma sekvensen.

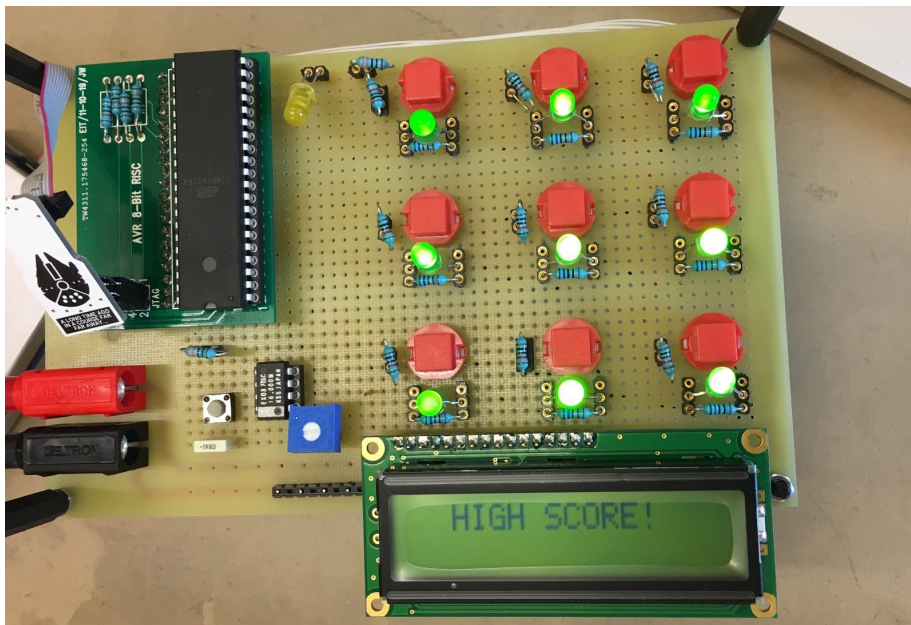


Bild 6 visar vad som händer vid high score.

## 7. Diskussion

### 7.1 Svårigheter

Under projektets gång har vi lärt oss mycket. Det har dock varit svårt att ta till sig information när man inte alls har mycket erfarenhet av varken ellära eller programmering i C eller assembler. För att använda information från datablad, laborationer och annat material behöver man lära sig vad det är man letar efter, hur det används, och man finner snabbt att man är på väg ner i ett kaninhål där man inte vet om man i slutet faktiskt kommer hitta det man behöver.

En stor del av projektets gång har vi tyvärr fastnat på grund av omständliga felsökningar. Framförallt lade vi mycket tid på att undersöka varför neopixlarna inte fungerade. Tyvärr kom varken vi eller handledare fram till orsaken till felet, och därmed fick vi göra en förenkling i projektet för att komma vidare.

Motgångar likt denna har gjort att vi flera gånger har behövt koppla om hårdvaran och uppdatera kopplingsschemat vilket skapat merjobb som inte behövts om vi vetat mer redan innan projektets början.

Sammanfattningsvis så har projektet haft sina svårigheter och motgångar som fördröjt arbetet och förenklat prototypen. Vi har trots detta uppnått den kravspecifikation vi skapade initialt och anser därför att projektet är lyckat.

## 8. Referenser

1. Datablad processor:  
<https://www.gme.cz/data/attachments/dsh.958-112.1.pdf>
2. LCD-display:  
<https://www.lawicel-shop.se/tillverkare/bolymin/lcd-2x16-backlight>
3. Lysdioder:  
[https://www.elfa.se/sv/led-5mm-rund-groen-40000mcd-520nm-rnd-components-rnd-135-00038/p/30100035?pos=2&origPos=3&origPageSize=10&prodprice=1.87775&q=led&p=cat-L2-3D\\_525341~cat-L3D\\_525297~cat-L2-3D\\_1914325~cat-DNAV\\_PL\\_020103&isProductFamily=false&campaign=&track=true](https://www.elfa.se/sv/led-5mm-rund-groen-40000mcd-520nm-rnd-components-rnd-135-00038/p/30100035?pos=2&origPos=3&origPageSize=10&prodprice=1.87775&q=led&p=cat-L2-3D_525341~cat-L3D_525297~cat-L2-3D_1914325~cat-DNAV_PL_020103&isProductFamily=false&campaign=&track=true)