

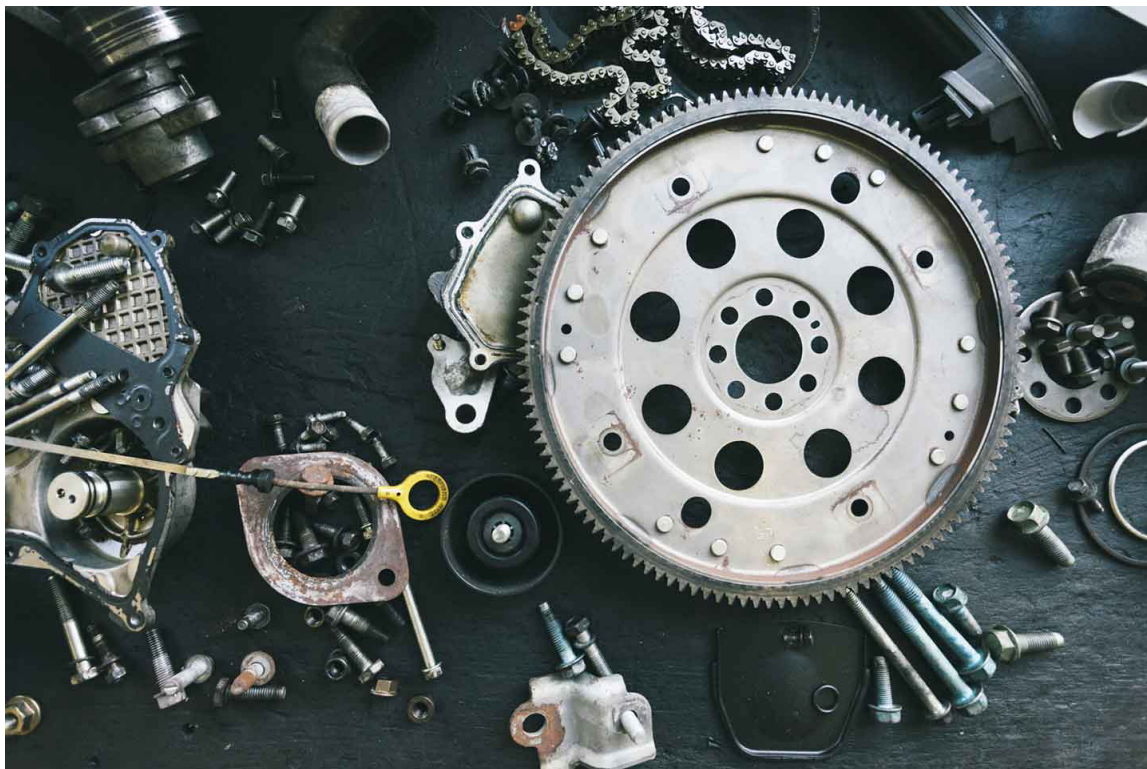
# THE MEMORY MASTER

*Av Fahad och Mumin*

## INTRODUKTION

---

The Memory master är ett spännande spel som testar och förbättrar dina minnesförmågor. Genom att upprepa allt längre sekvenser av blinkande LED-lampor kommer du att utmana ditt korttidsminne och dina reflexer. Med tre olika svårighetsnivåer är memory master ett spel för alla.



### Hur spelet fungerar

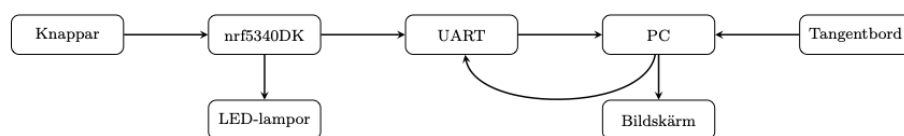
Din uppgift är att upprepa en växande sekvens av blinkande LED-lampor. Spelet börjar enkelt med en kort sekvens, men för varje runda du klarar blir spelet svårare

När spelet startar kommer en LED-lampa att blinka. Din uppgift är att trycka på motsvarande knapp. I nästa runda blinkar två LED-lampor i följd och du måste trycka på knapparna i samma ordning som lamporna blinkar. För varje runda du klarar läggs det ytterligare en LED till i sekvensen. Efter runda 4 ökar svårighetsgraden till "Medel" och efter runda 8 ökar svårighetsgraden till "Svårt" Detta betyder att lamporna lyser snabbare vilket gör spelet mer utmanande.

Du kan alltid få en ledtråd genom att klicka på "h" på tangentborden. Men ledtråden ger dig endast svaret till den första lampan.

Klarar du av alla tolv rundor har du väl bevisat att du är en riktigt Memory Master.

## Block diagram



## Hårdvarukomponenter:

**Nrf5340DK:** Kör spellogiken, genererar Led sekvenser samt läser knapptryckning.

**Knappar:** Spelarens inmatningsgränssnitt för att upprepa sekvenser.

**Led lampor:** Visar den sekvens spelaren ska memorera

**PC:** Visar spelstatus och tar emot ledtrådskommandon

Kommunikations komponenter:

**UART:** Kopplar nrf5340DK till PC för statusuppdateringar och kommandon.

## Kodimplementering

The Memory Master består av två huvudsakliga koder

1. Main.c - Som innehåller programmets startpunkt och initialiseringar
2. LEDgame.c/.h - Som implementerar spellogiken och alla hjälpfunktioner

## Flödesschema

Flödesschemat illustrerar hur memory master spelet fungerar från start till slut.

1. **Starta programmet:** Programmet startas
2. **Initiera system:** Systemet initialiseras, inklusive GPIO, UART, RTC och Leden.
3. **Starta spel (round = 0):** Spelet startas med round 0
4. **Kontroll (round < 12?):** Programmet kontrollerar om aktuell runda är mindre än 12  
  
Om ja: Spelet försätter till nästa steg  
Om Nej: Spelaren har klarat alla rundor och får meddelande att hen har vunnit
5. **Lägg till ny LED i sekvensen:** En ny slumpmässig LED läggs till i den befintliga sekvensen.
6. **Visa LED-sekvens:** Den aktuella sekvensen visas för spelaren genom att LED-lamporna blinkar i tur och ordning.

7. **Spelaren upprepar:** Spelaren försöker upprepa sekvensen genom att trycka på motsvarande knappar.
8. **Korrekt?:** Programmet kontrollerar om spelarens input matchar den korrekta sekvensen.  
Om Ja: Spelet fortsätter till nästa steg.  
Om Nej: "Game Over" visas och spelet avslutas.
9. Öka round: Rundnumret ökas med 1.
10. **Justera svårigheter:** Svårighetsgraden ökas baserat på rundnummer (snabbare LED-blink vid högre rundor).
11. **Återkoppling:** Processen återgår till kontrollen "round < 12?" för att antingen fortsätta med en ny runda eller avsluta spelet med vinst.

**Bild finns på nästa sida**

