

# Reaktionsspel - spels officiella dokumentation -

## Grupp 6 (Butros Nasrallah & Nabih Shankan)

Det här programmet skapar ett enkelt spel där snabb reaktion testas på en nRF5340 UARTE kretskort. Det använder UART för att prata med användaren, GPIO för att styra en LED och knapp, samt SysTick för att mäta tiden. RTC ger en slumpmässig startfördröjning, så spelet börjar vid olika tidpunkter.

### Komponenter som används:

- LED (PIN 28): Visar signalen när spelaren ska trycka knappen.
- Knapp (BUTTON1): Användaren trycker här för att vinna när ledden lyser.
- UART (TXD: 20, RXD: 22): Skickar och tar emot meddelanden till Putty samt tar ifrån puttytyn också.

### Programflöde för play\_reaction\_game

1. UART skickar instruktioner till användaren.
2. Programmet väntar en slumpmässig tid (2–5 sekunder).
3. LED tänds.
4. Tiden startar.
5. Väntar på knapptryck t.ex (3 sekunder).
6. LED släcks.
7. Om knappen trycks i tid så visar meddelandet "Du vann!" och om inte annars "För långsam...".
8. Användaren trycker trycker Enter för att spela igen efter begäran från Puttytyn.

### Pseudokod för spelets huvudfunktion:

```
FUNKTION play_reaction_game:  
SKRIV "Tryck på knappen när LED lyser!" UARTE  
ANROPA set_random_seed()  
SLUMPA ett heltal mellan 2000 och 5000 (ms)  
VÄNTA  
TÄND LED1  
SÄTT button_pressed TILL 0  
SPARA starttid från SysTick  
WHILE knapp trycks inte OCH tiden < 3 sekunder:
```

```
IF button_pressed ÄR 1:  
BRYT  
IF 3 sekunder HAR GÅTT:  
BRYT  
SLÄCK LED1  
RENSA skärm  
IF button_pressed ÄR 1:  
SKRIV "Du VANN!" UARTE  
ELSE SKRIV "För långsam. Du FÖRLORADE." UARTE  
SKRIV "Spela igen? Tryck på Enter..." UARTE  
VÄNTA på input via read_string  
SLUT FUNKTION
```

## Styrkor i implementationen

- UART gör det enkelt att skicka och ta emot information, vilket ger den önskade kommunikationslösning.
- RTC används för att skapa en slumpmässig men logisk eller med andra ord realistiskt väntetid, vilket gör att systemet känns enkelt och möjligt att spela.
- SysTick används för att mäta tid med hög fokus, vilket säkrar exakta tidsberäkningar i systemet under och efter spelet.