

X si 0 NUCELO STM32F446RE

Boticean Stefan-Andrei

Jurca Robert

Introducere

Acest proiect implementeaza jocul X si 0 pe un microcontroler STM32 utilizand comunicarea UART si un LED pentru interactiune.

Configurare Hardware

In cadrul acestui proiect am utilizat placa Nucelo STM32F446RE.

Configurarea hardware implica pregatirea microcontrolerului STM32 pentru a interactiona cu utilizatorul si a afisa semnale de stare.

1. LED-ul Onboard:

- LED-ul onboard este conectat la pinul PA5 al microcontrolerului STM32.
- Functia “LED_Init()” configureaza pinul PA5 ca iesire. Acest lucru este realizat activand clock-ul pentru GPIOA si setand modul pinului PA5 la output.
- LED-ul este folosit pentru a indica activitatile de citire si scriere prin UART. Functiile “LED_On()” si “LED_Off()” controleaza starea LED-ului, iar “LED_Flash()” il face sa clipeasca pentru a semnala o actiune.

2. UART pentru Comunicarea Seriala:

- UART2 este folosit pentru comunicarea seriala cu PC-ul sau alte dispozitive.
- Functia “UART2_Init()” configureaza UART2 pentru comunicare la baud rate-ul de 115200 bps.

- Pini PA2 si PA3 sunt configurati ca functii alternative pentru TX si RX.

- Baud rate-ul este setat la 115200 bps pentru un clock de 16 MHz.

- Functiile “UART2_Write()” si “UART2_Read()” gestioneaza transmiterea si primirea datelor prin UART.

Functionarea generala a codului

Codul implementeaza logica jocului X si 0 si interactioneaza cu utilizatorul prin UART, folosind LED-ul pentru a semnaliza operatiunile de citire si scriere. Mai jos este o descriere detaliata a functionarii generale a codului:

1. Initializarea sistemului:

- “SystemInit()”: Initializeaza sistemul.
- “UART2_Init()”: Configureaza UART2 pentru comunicare seriala.
- “LED_Init()”: Configureaza LED-ul onboard pentru a indica activitatea.

2. Interactiunea cu utilizatorul:

- Codul cere utilizatorului sa aleaga un simbol (X sau O).
- In functie de alegerea utilizatorului, simbolul computerului este setat la celalalt simbol.

3. Initializarea placii de joc:

- “initializeBoard()”: Initializeaza placa de joc cu caractere "-".

4. Bucla principala a jocului:

- Placa de joc este afisata folosind “displayBoard()”.
- Utilizatorul este rugat sa introduca coordonatele pentru mutare.

- Dupa fiecare mutare, se verifica daca utilizatorul sau computerul a castigat folosind “checkWin()”.
- Daca nu exista un castigator, se verifica daca jocul este remiza folosind “checkDraw()”.
- Computerul face o mutare aleatorie daca jocul nu s-a terminat.

5. Incheierea jocului:

- Daca utilizatorul sau computerul castiga, jocul se incheie si placa de joc este afisata.
- Daca jocul este remiza, aceasta este afisata si jocul se incheie.

LED-ul este folosit pentru a semnaliza activitatile de citire si scriere prin UART, clipind de fiecare data cand datele sunt transmise sau primite.

Detalii despre cod

Link cod:

https://github.com/BoticeanStefan/SI_proiect/blob/main/LED_main

Functii principale:

- initializeBoard: Initializeaza tabla de joc cu caractere "-".
- displayBoard: Afiseaza tabla de joc.
- checkWin: Verifica daca un jucator a castigat.
- checkDraw: Verifica daca jocul este remiza.
- void LED_Init(void)

Aceasta functie initializeaza LED-ul onboard.

1. RCC->AHB1ENR |= 0x01; - Activeaza ceasul pentru GPIOA.

2. `GPIOA->MODER |= (1 << (LED_PIN * 2));` - Seteaza pinul PA5 ca iesire.

- `void LED_On(void)`

Aceasta functie aprinde LED-ul.

1. `GPIOA->BSRR = (1 << LED_PIN);` - Seteaza pinul PA5 la HIGH.

- `void LED_Off(void)`

Aceasta functie stinge LED-ul.

1. `GPIOA->BSRR = (1 << (LED_PIN + 16));` - Reseteaza pinul PA5 la LOW.

- `void LED_Flash(void)`

Aceasta functie face LED-ul sa clipeasca.

1. `LED_On();` - Aprinde LED-ul.

2. `for (volatile int i = 0; i < 100000; i++);` - Introduce o intarziere simpla.

3. `LED_Off();` - Stinge LED-ul.

4. `for (volatile int i = 0; i < 100000; i++);` - Introduce o alta intarziere.

- `void UART2_Init(void)`

Aceasta functie initializeaza UART2 pentru comunicare seriala.

1. `RCC->APB1ENR |= 0x20000;` - Activeaza ceasul pentru UART2.

2. `RCC->AHB1ENR |= 0x01;` - Activeaza ceasul pentru GPIOA.

3. `GPIOA->MODER &= ~0x00F0;` - Configureaza PA2 si PA3 ca functie alternativa.

4. `GPIOA->MODER |= 0x00A0;` - Configureaza PA2 si PA3 ca functie alternativa.

5. `GPIOA->AFR[0] &= ~0xFF00;` - Selecteaza functia alternativa AF7 pentru PA2 si PA3.

6. GPIOA->AFR[0] |= 0x7700; - Selecteaza functia alternativa AF7 pentru PA2 si PA3.
7. USART2->BRR = 0x008B; - Configureaza baud rate-ul la 115200 pentru 16 MHz.
8. USART2->CR1 = 0x000C; - Activeaza transmitatorul si receptorul.
9. USART2->CR1 |= 0x2000; - Activeaza UART-ul.

- void UART2_Write(int ch)

Aceasta functie scrie un caracter prin UART2.

1. while (!(USART2->SR & USART_SR_TXE)) { } - Asteapta pana bufferul de transmitere este gol.
2. USART2->DR = (ch & 0xFF); - Trimite caracterul prin registrul de date.
3. LED_Flash(); - Face LED-ul sa clipeasca pentru a semnala transmiterea.

- int UART2_Read(void)

Aceasta functie citeste un caracter prin UART2.

1. while (!(USART2->SR & USART_SR_RXNE)) { } - Asteapta pana bufferul de primire nu este gol.
2. LED_Flash(); - Face LED-ul sa clipeasca pentru a semnala primirea.
3. return USART2->DR; - Returneaza caracterul citit.

- int fputc(int ch, FILE *f)

Aceasta functie scrie un caracter folosind functia UART2_Write.

1. UART2_Write(ch); - Scrie caracterul folosind UART2.
2. return ch; - Returneaza caracterul scris.

- int fgetc(FILE *f)

Aceasta functie citeste un caracter folosind functia UART2_Read.

1. return UART2_Read(); - Citeste si returneaza caracterul folosind UART2.

Gameplay

In terminalul aplicatiei TeraTerm, apare mesajul “Choose your mark” prin care utilizatorul este rugat sa isi aleaga simbolul X sau 0 dupa care table de joc este afisata. Apoi utilizatorul este rugat sa introduca linia si coloana unde doreste sa faca mutarea. Computerul face mutari aleatorii, iar jocul se finalizeaza cand unul dintre jucatori castiga sau este remiza.

Concluzii

Acest proiect demonstreaza utilizarea STM32 pentru un joc simplu de X si 0 si interactiunea prin UART cu terminalul TeraTerm.

Posibile imbunatatiri includ algoritmi mai avansati pentru mutarile computerului.

```
VT COM10 - Tera Term VT
File Edit Setup Control Window Help
Choose your mark (X or O):
  0 1 2
0 - - -
1 - - -
2 - - -
Enter row and column (0-2) for your move:
```

Flow chart



