



Rapport

Seare Taklab – Seare.taklab@yh.nackdemin.se

Innehållsförteckning

Introduktion	3
Beskrivning	4
Slutsats	6
Github-Länk	7

Introduktion

Den här projekten är inriktad på utvecklingen av en UART-drivrutin för STM32F411x-plattformen. Målet är att konstruera en driftsäker och effektiv rutin som hanterar kommunikationen mellan mikrokontrollen och andra enheter som använder UART-protokollet. I denna Rapport ger jag en förtydligat beskrivning av designprocessen, inkluderat implementeringen på plattformen. Jag utför tester och utvärderingar av rutinen för att säkerställa dess pålitlighet och effektivitet. Dessutom presenterar jag plattformens egenskaper och begränsningar samt ger en översikt över olika komponenter i rutinen. Rapporten inkluderar även implementeringen av styrningen för 3 LED-lampor som är kopplade till mikrokontrollen och kan styras för att slås av och på.

Beskrivning

I filen "Led.cpp" implementeras medlemsfunktionerna för LED-klassen. Konstruktorn

"Led::Led()" används för att initialisera LED-lamporna med en given färg och status.

Beroende på färg och status väljs den motsvarande porten och LED-pinnarna konfigureras.

Medlemsfunktionen "Led::setState()" används för att ändra statusen för en LED. Beroende på statusen slås LED-pinnen antingen på eller av. Funktionen "Led::getState()" returnerar den aktuella statusen för en LED-lampa.

I "Led.h"-filen definieras LED-klassen med sina egenskaper och medlemsfunktioner för att hantera LED-lampan. Egenskaperna "color" och "state" representerar färgen och statusen för lampan. Konstruktorn tar emot information om färgen och statusen och tilldelar dessa värden till de motsvarande egenskaperna. För att ändra lampan status används "setState()", medan "getState()" används för att hämta den nuvarande statusen. Dessa medlemsfunktioner ger möjlighet att styra och övervaka LED-lampan inom LED-klassen.

I "Uart.h"-filen finns deklarationerna för funktionerna "USART2_Init()" och "test_setup()".

Funktionen "USART2_Init()" används för att initiera UART-protokollet och dess komponenter. Det innefattar konfiguration av klocksignaler och GPIO-portar för att möjliggöra UART-kommunikation.

I "Uart.cpp"-filen implementeras funktionerna "USART2_Init()", "USART2_write()" och "USART2_read()". Funktionen "USART2_Init()" är ansvarig för att initiera UART2 genom att aktivera klocktillgång och konfigurera GPIO-pinnarna för att använda den alternativa UART-kommunikationsfunktionen. Baudhastighet och andra överföringsparametrar ställs också in

för UART2. Dessa åtgärder säkerställer att UART2 är korrekt konfigurerad och redo för seriell kommunikation. "USART2_write()" används för att skicka information till terminalen via UART när överföringsregistret är tomt och ny data ska skickas till dataregistret. "USART2_read()" läser data från UART-mottagaren och väntar på att den lästa datan ska skickas tillbaka.

I "Main.cpp"-filen finns huvudfunktionen "main()", som hanterar LED-lamporna och UART-kommunikationen. Först initieras UART-kommunikationen med hjälp av funktionen "USART2_Init()". Sedan skapas tre LED-objekt: "led1" och "led2" som lokala variabler, och "led3" som skapas dynamiskt med hjälp av "new". "led1" har färgen RÖD och statusen PÅ, "led2" har färgen BLÅ och statusen PÅ, och "led3" har färgen GUL och statusen PÅ.

Därefter hämtas och sparas statusen för "led1" i variabeln "

Slutsats

Detta projekt sammanför STM32 mikrokontroller och UART-kommunikation för att skapa en flexibel lösning för styrning av tre LED-lampor. Applikationen har en bred användning inom områden som IoT, hemautomation, elektronikprototyper och belysningsdesign, med mera. Genom att arbeta med STM32-plattformen har jag inte bara fått praktisk erfarenhet av att implementera UART-kommunikation, utan även utökat min kunskap inom detta område. Detta projekt har varit en värdefull inlärningsupplevelse och jag ser fram emot att fortsätta utforska och vidareutveckla mina färdigheter inom detta spännande område.

Github-Länk

https://github.com/Seta2022/Inl-mingsuppgift_UART