

Mjukvaruutvecklare, Inbyggda system och IoT

Mahadi Hasan

[Mahadi.Hasan@yh.nackademin.se](mailto:Mahadi.Hasan@yh.nackademin.se)

Hantera en LCD med UART-protokollet

2023

Innehållsförteckning

[Abstrakt 2](#_Toc133582195)

[Introduktion 2](#_Toc133582196)

[Bakgrund 3](#_Toc133582197)

[STM32 Arkitektur för mikrokontroller 4](#_Toc133582198)

[Metodik: 5](#_Toc133582199)

[Slutsats: 6](#_Toc133582200)

# Abstrakt

Denna rapport diskuterar processen för att styra Liquid Crystal Display (LCD) med hjälp av en STM32-mikrokontroller. STM32-mikrokontrollern är ett populärt val för design av inbyggda system på grund av dess låga strömförbrukning, höga processorkraft och flexibilitet. I denna rapport, vi presenterar de grundläggande principerna för LCD-drift, STM32-mikrokontrollerarkitekturen, och processen för gränssnitt mellan LCD och STM32-mikrokontrollern. Vi presenterar också källkod i källkodsmappen som visar hur man styr LCD med STM32-mikrokontrollern.

# Introduktion

LCD-teknik (Liquid Crystal Display) används ofta i elektroniska enheter som utgångsdisplay. LCD-skärmar kan hanteras med olika protokoll. En LCD-skärm (Liquid Crystal Display) med UART-protokoll (Universal Asynchronous Receiver Transmitter) är en typ av LCD-skärm som kan hanteras med UART-kommunikationsprotokollet. Detta innebär att LCD-skärmen kan ta emot data och kommandon från en mikrokontroll eller annan enhet som använder UART-kommunikation.

För att använda en LCD med UART-protokoll måste mikrokontrollern upprätta en kommunikationslänk med LCD-skärmen genom att ansluta TX- och RX-stiften på mikrokontrollern till motsvarande stift på LCD-skärmen. UART-modulen för mikrokontrollern måste också initieras genom att ställa in baudhastighet, paritet och andra parametrar.

När kommunikationslänken har upprättats och UART-modulen har initierats kan mikrokontrollern skicka kommandon och data till LCD-skärmen med UART-protokollet. Kommandona kan användas för att styra skärmen, till exempel att slå på / av, ställa in markörens position och rensa skärmen. Data kan skickas till LCD-skärmen för att visa text, bilder och annat grafiskt innehåll.

Fördelen med att använda en LCD med UART-protokoll är att det förenklar processen att hantera skärmen. I stället för att använda komplexa hårdvarugränssnitt kan mikrokontrollern kommunicera med LCD-skärmen med ett enkelt UART-gränssnitt. Detta gör det lättare att styra skärmen och minskar de hårdvaruresurser som krävs.

En LCD med UART-protokoll är ett användbart verktyg för att visa utdata i elektroniska enheter som använder mikrokontroller. Det ger ett enkelt och effektivt sätt att hantera displayen, vilket kan spara tid och resurser vid design och implementering av elektroniska system.

# Bakgrund

UART (Universal Asynchronous Receiver-Transmitter) är ett kommunikationsprotokoll som används för seriell kommunikation mellan två enheter. Det är ett enkelt och allmänt använt protokoll som möjliggör dataöverföring mellan enheter med endast två ledningar (RX och TX).

UART fungerar asynkront, vilket innebär att data överförs utan användning av en extern klocksignal. I stället måste de sändande och mottagande enheterna i förväg komma överens om kommunikationshastigheten, som vanligtvis mäts i baudhastighet.

I UART-kommunikation överförs data som en serie bitar, där varje byte av data inramas av en start bit, en eller flera databitar, en valfri paritetsbit (för felkontroll) och en stoppbit. Start biten är alltid låg (0) och stoppbiten är alltid hög (1), med databitarna och paritetsbiten (om de används) placerade däremellan.

En LCD-skärm är en enhet som vanligtvis används som utgångsdisplay i elektroniska enheter. LCD-skärmen kan visa text, bilder och annat grafiskt innehåll. För att hantera en LCD med UART-protokollet måste vi upprätta en kommunikationslänk mellan mikrokontrollern och LCD-skärmen.

# STM32 Arkitektur för mikrokontroller

STM32-mikrokontrollerarkitekturen är baserad på ARM Cortex-M-processorarkitekturen, som är en populär arkitektur för inbyggda system på grund av dess låga strömförbrukning, höga prestanda och användarvänlighet. STM32-mikrokontrollerfamiljen stöder olika ARM Cortex-M-processorkärnor, inklusive Cortex-M0, Cortex-M0+, Cortex-M3, Cortex-M4, Cortex-M7 och Cortex-M23 / M33.

STM32-mikrokontrollerna har vanligtvis följande nyckelkomponenter:

1. Kärna: Detta är centralenheten (CPU) som utför instruktioner och utför beräkningar. Cortex-M-kärnorna som används i STM32-mikrokontroller är 32-bitars RISC-processorer (Reduced Instruction Set Computer) som erbjuder hög prestanda och låg strömförbrukning.

2. Minne: STM32-mikrokontrollerna har vanligtvis minne på chipet, inklusive flashminne för lagring av programkoden och RAM (Random Access Memory) för lagring av data och stack.

3. Kringutrustning: STM32-mikrokontrollerna har ett brett utbud av kringutrustning, inklusive GPIO, timers, ADC, DAC, USART, SPI, I2Cs, USB, CAN, Ethernet och mer. Dessa kringutrustning ger funktionalitet för att samverka med den externa världen och kommunicera med andra enheter.

4. Klock- och strömhantering: STM32-mikrokontrollerna har ett sofistikerat klock- och strömhanteringssystem som möjliggör effektiv drift och strömförbrukning. Klocksystemet genererar olika klocksignaler som krävs för drift av mikrokontroller och kringutrustning. Strömhanteringssystemet gör att mikrokontrollern kan gå in i lågeffektlägen när den inte används, vilket minskar strömförbrukningen.

5. Felsökning och programmering: STM32-mikrokontrollerna stöder vanligtvis olika felsöknings- och programmeringsgränssnitt, inklusive JTAG, SWD (Serial Wire Debug) och BOOT (Bootloader) lägen.

Sammantaget ger STM32-mikrokontrollerarkitekturen en kraftfull och flexibel plattform för att utveckla inbyggda system med ett brett utbud av funktioner och funktioner. Arkitekturen stöder en rik uppsättning kringutrustning, låg strömförbrukning, effektiv drift och användarvänlighet, vilket gör den till ett idealiskt val för ett brett spektrum av applikationer.

# Metodik:

För att hantera en LCD med UART-protokollet måste vi följa följande steg:

Steg 1: Upprätta en kommunikationslänk mellan mikrokontrollern och LCD-skärmen. Detta kan göras genom att ansluta TX- och RX-stiften på mikrokontrollern till motsvarande stift på LCD-skärmen.

Steg 2: Initiera mikrokontrollerns UART-modul. Detta kan göras genom att ställa in baudhastighet, paritet och andra parametrar för UART-modulen.

Steg 3: Implementera en funktion som kan ta emot och tolka UART-kommandon och utföra lämpliga åtgärder på LCD-skärmen baserat på det mottagna kommandot. Detta kan innebära att skicka specifika bytesekvenser till LCD-skärmen för att initiera önskade åtgärder, till exempel att ställa in markörens position eller skriva data till skärmen.

Steg 4: Implementera en funktion som kan användas för att skicka data till LCD-skärmen via UART, till exempel textsträngar eller annan visningsinformation. Denna funktion bör ta hand om att bryta ner data i lämpliga bytesekvenser och skicka dem till LCD-skärmen i rätt ordning.

Steg 5: Kontrollera LCD-skärmen. Detta kan göras genom att skicka lämpliga kontrollkommandon till LCD-skärmen med UART-protokollet. Kontrollkommandona kan användas för att slå på / stänga av skärmen, ställa in markörens position och rensa displayen.

Steg 6: Implementera felhanteringskod för att hantera fall där UART-kommunikationen störs eller LCD-skärmen inte svarar korrekt på kommandon. Detta kan omfatta att försöka igen misslyckade kommandon, rapportera fel tillbaka till användaren eller vidta andra lämpliga åtgärder efter behov.

# Slutsats:

Sammanfattningsvis är hantering av en LCD med UART-protokollet ett enkelt och effektivt sätt att visa utdata på en LCD-skärm. UART-protokollet används ofta i mikrokontroller, och LCD-skärmen används ofta som utgångsdisplay i elektroniska enheter. Genom att följa stegen som beskrivs i denna rapport kan vi framgångsrikt hantera en LCD-skärm med UART-protokollet.