

Loggningsfunktion för HKM-robot

Sammanfattning

Detta examensarbete fokuserade på att utveckla en loggningsfunktion för PLC-programmering med Structured Text för HKM-robotar. Syftet var att skapa en användarvänlig lösning för loggning av meddelanden på olika nivåer (Trace, Debug, Info, Error och Warn) för att underlätta övervakning och felsökning av PLC-program. Genom en systematisk metod och analys av befintliga loggfunktioner samt studier av SLF4J-konceptet utvecklades loggningsfunktionen stegvis. Testresultaten visade att loggningsfunktionen effektivt loggade meddelanden till debug output trace och filer. Den utvecklade loggningsfunktionen förbättrar övervakningen och felsökningen av PLC-koder och kan vidareutvecklas för framtida behov.

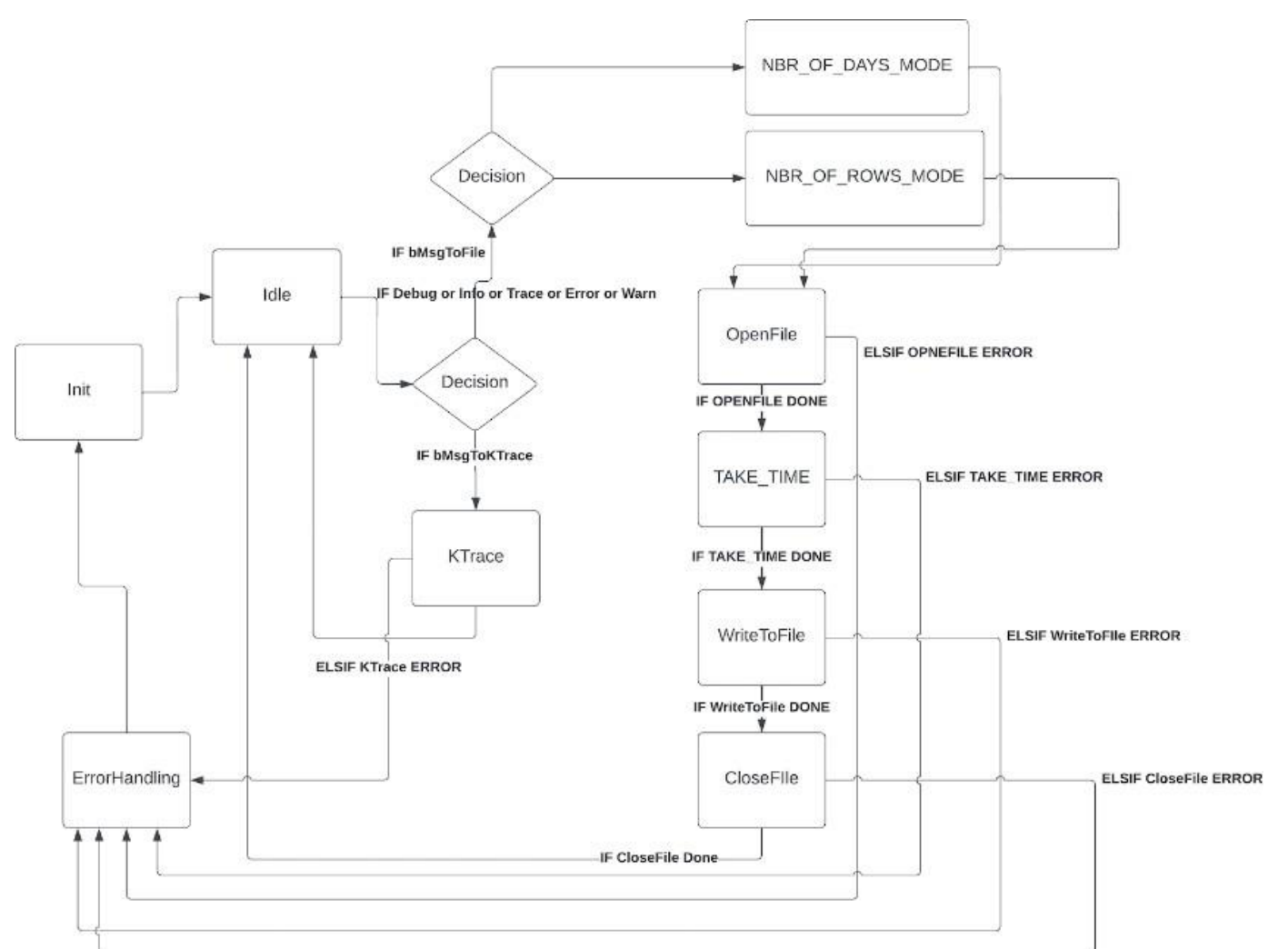
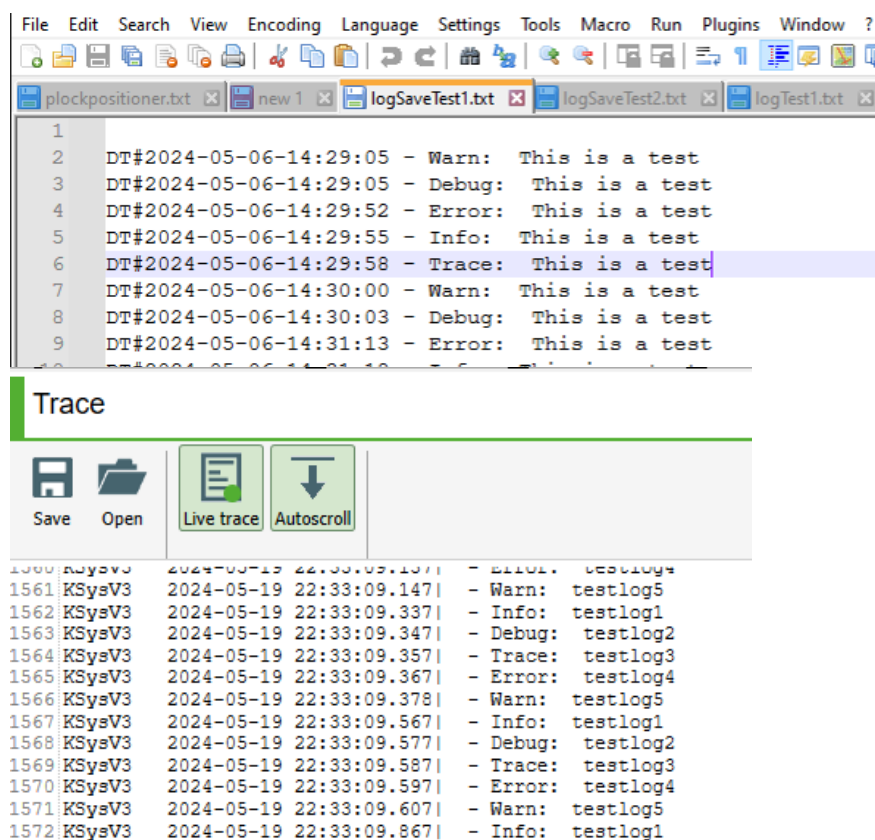
Problemformulering

- Hur kan befintlig loggfunktionalitet förbättras för att möjliggöra enklare felsökning och underhåll av HKM-robotens PLC-kod?
- Hur kan kodspårning implementeras för att lokalisera anrop i källkoden och underlätta identifiering av problem.
- Vilka flaggor som kontrollerar affinitets- och allvarlighetsgrad är relevanta för att möta användarens behov av att kontrollera loggnivån?
- Hur kan den utökade loggfunktionaliteten göras återanvändbar som en bibliotekskomponent för PLC-programmerare?

Metod

För att utveckla en loggningsfunktion för HKM-robot användes en systematisk metodik. Först analyserades befintliga loggningssystem och SLF4J-konceptet för att förstå de bästa praxis för loggning. Därefter utvecklades loggningsfunktionen stegvis med Structured Text, en programmeringsspråk som används inom PLC. Koden för loggfunktionen har utgått utifrån sekventiell programmering med modularitet och dessa moduler är sedan sammankopplade för att bygga upp till en hel loggningsfunktion. Funktionen testades genom att köra den under olika scenarier och logga meddelanden på olika nivåer (Trace, Debug, Info, Error och Warn). Under prestandamätningar mättes exekveringstider för olika olika delar av koden.

Implementering

```

1
2 DT#2024-05-06-14:29:05 - Warn: This is a test
3 DT#2024-05-06-14:29:05 - Debug: This is a test
4 DT#2024-05-06-14:29:52 - Error: This is a test
5 DT#2024-05-06-14:29:55 - Info: This is a test
6 DT#2024-05-06-14:29:58 - Trace: This is a test
7 DT#2024-05-06-14:30:00 - Warn: This is a test
8 DT#2024-05-06-14:30:03 - Debug: This is a test
9 DT#2024-05-06-14:31:13 - Error: This is a test

```

Resultat

Loggningsfunktionen presterade som förväntat under testningen och visade sig vara kapabel att hantera loggmeddelanden på olika nivåer (Trace, Info, Debug, Warn, Error) i både debug output trace och filer.

Prestandatestningen visade att tiden för att skriva loggmeddelanden ökade proportionellt med antalet meddelanden. Specifikt ökade tiden för tillståndet KTRACE med 10 ms och WRITE_TO_FILE med 20 ms i genomsnitt för varje extra loggmeddelande.

Loggningsfunktionen klarade att skriva många loggmeddelanden i rad så länge vektorer för loggmeddelanden inte överskred sina gränser på 1000 strängar. Överskridande av denna gräns resulterade i att loggningsfunktionen slutade fungera och visade ett exception error. Användarupplevelsen under testning med HKM-roboten var positiv, med loggningsfunktionen som både användarvänlig och lätt att integrera i PLC-programmet. Funktionen möjliggjorde samtidig live-spårning och loggspårning utan att påverka robotens prestanda när den kördes i en egen tråd i PLC.

Slutsats

Den förbättrade loggningsfunktionen är pålitlig och effektiv för felsökning och underhåll av HKM-robotens PLC-kod. Denna loggningsfunktion kan återanvändas som en bibliotekskomponent för PLC-programmerare, vilket bidrar till enklare felsökning i framtida projekt. Begränsningen på 1000 loggmeddelanden per vektor måste dock beaktas. Sammanfattningsvis har loggningsfunktionen förbättrat användarupplevelsen och underlättat kodspårning.