

Kapittel 2

Brukergrensesnittet i LabVIEW

2.1 Hvordan starte LabVIEW

Programmet LabVIEW kan startes på flere måter:

- Via Start (på PC'ens skrivebord) / Programmer / National Instruments / LabVIEW

Du kan åpne LabVIEW på denne måten nå.

- Ved å klikke på LabVIEW-ikonet på PC'ens skrivebord (dersom det fins et ikon der, da – det er nemlig ikke sikkert at det er en snarvei via et slikt ikon til LabVIEW selv om LabVIEW er installert).
- Ved å åpne en eksisterende LabVIEW-fil (dobbelklikke på fil-ikonet) i Windows Utforsker. Slike filer har et LabVIEW-ikon som fil-ikon, og filene har filnavn ***.vi**.

2.2 Oppstartsvinduet

Figur 2.1 viser LabVIEWs oppstartsvindu.

Kommentarer til de enkelte knappene i oppstartsvinduet:

- **New** åpner en ny VI.
- **Open** åpner en eksisterende VI. Med pilknappen til høyre for Open-knappen får du opp en liste over de sist åpnede filene.



Figur 2.1: LabVIEWs oppstartsvindu

- **DAQ Solutions**¹ gir tilgang til ferdige VI'er for forskjellige anvendelser av LabVIEW til måling (datainnsamling) og styring, f.eks. innlesning av spenning og strøm, innlesning av temperaturmålinger fra termoelementer og regulering av prosesser med PID-regulator. Disse VI'ene er basert på National Instruments eget I/O-utstyr, men de kan benyttes som utgangspunkt for utvikling av VI'er der en benytter I/O-utstyr fra andre leverandører.
- **Search Examples** gir tilgang til en stor samling av ferdigprogrammerte VI'er. Enkelte av disse VI'ene (som fins under "LabVIEW Fundamentals") demonstrerer ulike *muligheter* i LabVIEW som programmeringsspråk, mens andre demonstrerer *anvendelser* av LabVIEW, f.eks. for filtrering, matematisk analyse og regulering. Hver av disse VI'ene kan du kopiere og modifisere til eget bruk.
- **LabVIEW Tutorial** åpner et opplæringsprogram, som bl.a. inneholder et sett av aktiviteter.

2.3 VI'er - bruk og oppbygning

Vi skal som eksempel studere VI'en Temperature System Demo.vi, som følger med LabVIEW. Vi kan åpne denne VI'en via knappen Search Examples i LabVIEWs oppstartsvindu (eller via menyvalget Help /

¹DAQ = Data Acquisition = datainnsamling

Examples i en allerede åpnet VI). VI'en fins som Search (eller Find) Examples / Industry Applications / Analysis / Temperature System Demo.vi.

Når VI'en åpnes i LabVIEW, åpnes egentlig *to* ulike vinduer – ett for VI'ens *frontpanel* og ett for dens *diagram*:

- **Frontpanelet** (eng.: Front Panel) viser *grensesnittet som VI'en har mot brukeren*. Dette grensesnittet utgjøres av skruknapper (potmeter), brytere, grafer, tallfelt, skyvere, trykknapper osv.
- **Diagrammet** (eng.: Diagram) viser programmet som realiserer *VI'ens funksjonalitet*. Diagrammet uttrykker hvordan elementene på frontpanelet er koplet sammen funksjonsmessig. Elementene på frontpanelet vil bli representert med *terminaler* i diagrammet, og terminalene har ikoner med farge og tekst som skal uttrykke hva slags *datatype* som terminalene representerer. Vi kan si at *terminalene er variable* i programmet. De ulike datatypene i LabVIEW beskrives i kapittel 5.

Frontpanelet for Temperature System Demo.vi er vist i figur 2.2, og VI'ens diagram er vist i figur 2.3.

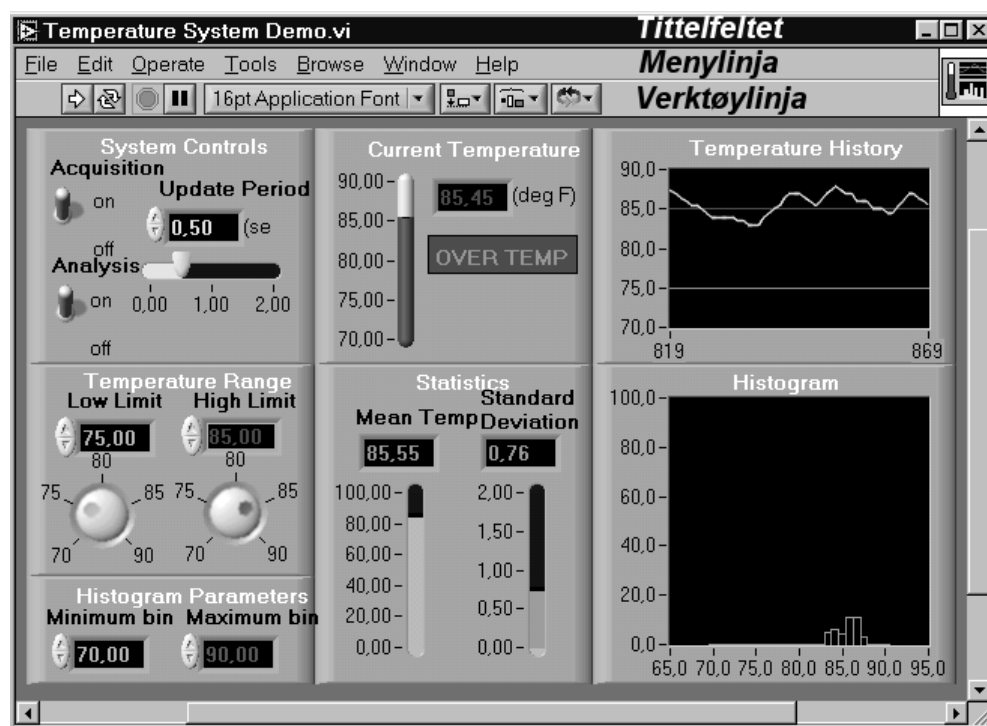
En VI kan sammenliknes med et elektronisk instrument: VI'ens frontpanel svarer til instrumentets frontpanel med brytere, tallfelter osv., og VI'ens diagram svarer til elektronikken inne i instrumentboksen (instrumentets funksjonalitet er realisert i disse elektroniske kretsene). Vi skal straks studere frontpanelet og diagrammet for Temperature System Demo.vi, men først *kjør* vi denne VI'en.

I VI'ens *verktøylinje*, se figur 2.2, er det knapper for å starte og stoppe en VI:

- **Start, og normal stopp:** Du starter en VI ved å trykke Run-knappen. Når VI'en har kjørt ferdig, dvs. når programmer er utført, stopper VI'en av seg selv. Noen VI'er lages slik at de skal utføre en oppgave bare én gang og deretter stoppe straks, mens andre VI'er lages slik at de utfører en oppgave om igjen og om igjen, helt til brukeren har trykket på en stoppknapp. Sistnevnte VI'er realiseres med en såkalt while-løkke, jf. kap. 3.5.3.

Temperature System Demo.vi går kontinuerlig, og den skal (normalt) stoppes vha. Aquisition-bryteren til venstre i frontpanelet, se figur 2.2.

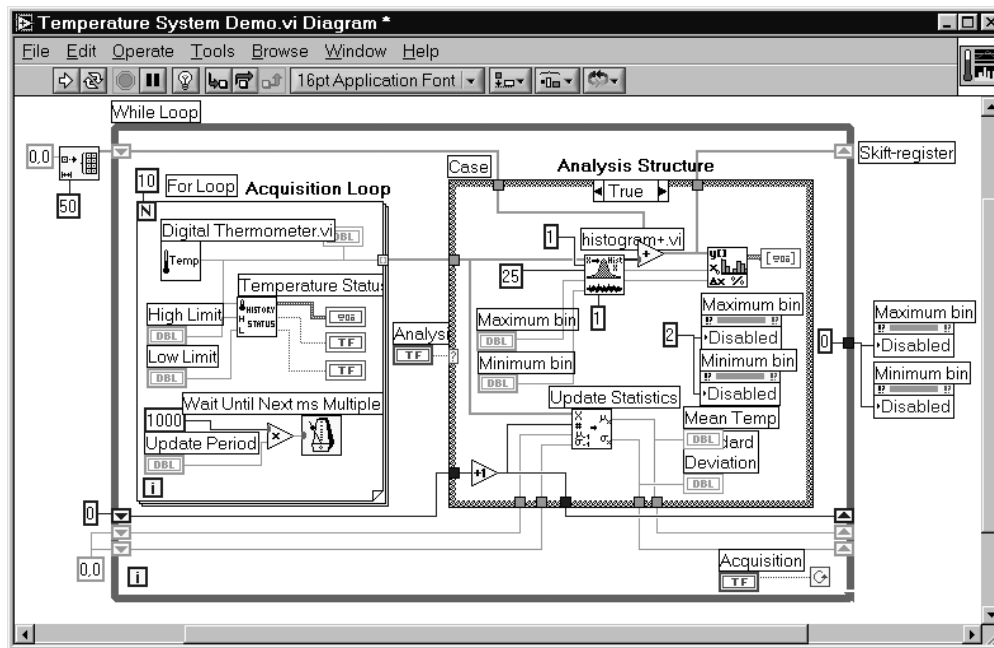
- **Tvangsstopp:** Du kan tvangsstoppe en VI som kjører, med Abort-knappen. Abort-knappen har en rød sirkel som symbol.



Figur 2.2: Frontpanelet for Temperature System Demo.vi

Merk: Dersom du som programutvikler ønsker at brukeren skal kunne stoppe VI'en, bør du ikke legge opp til at brukeren skal bruke Abort-knappen til dette. Bruk av Abort-knappen innebærer nemlig at normale stopp-prosedyrer ikke blir fulgt, og dette kan gjøre at data ikke lagres korrekt når VI'en avsluttes. I stedet bør du (som programutvikler) legge inn muligheten til å foreta en *kontrollert* stopp, vanligvis ved å legge inn en stoppknapp knyttet til en while-løkke, jf. kap. 3.5.3.

- **Repetert kjøring:** Du kan foreta repetert kjøring av en VI med Run Continuously-knappen, som er til høyre for Run-knappen. Merk at dette innebærer at VI'en starter, stopper, starter, stopper osv. Dette er *ikke den normale måten* å sørge for at en VI går kontinuerlig på, f.eks. i forbindelse med datainnsamling eller styring. I slike tilfeller er skal du heller benytte en while-løkke, jf. kap. 3.5.3. Repetert kjøring er aktuelt ved testing av en VI når du ønsker å se om VI'en fungerer riktig under varierte betingelser (du kan f.eks justere innstillingen av en skruknapp på frontpanelet under den repeterte kjøringen).
- **Pause:** Du kan sette VI'en i pausemodus med Pause-knappen.



Figur 2.3: Diagrammet for Temperature System Demo.vi

Pause-knappen har to tykke vertikale streker som symbol.

Vi skal nå studere frontpanelet og diagrammet for Temperature System Demo.vi. I den følgende teksten står LabVIEW-terminologi i *kursiv*.

Vi ser først på *frontpanelet*, som er vist i figur 2.2:

- Det aller øverste feltet (blått) i frontpanelvinduet er VI'ens *tittelfelt* (eng.: title bar).
- Under tittelfeltet er *menylinja* (menu bar).
- Under menylinja er *verktøylinja* (toolbar).
- Selve frontpanelet inneholder elementer som brytere, tallfelter, søyleindikatorer og grafer, som utgjør VI'ens brukergrensesnitt.
- Elementene i feltene i frontpanelet kalt Systems Controls, Histogram Parameters og Temperature Range kan betegnes *inngangselementer* (eng.: controls), som brukeren kan manipulere mens VI'en kjører.
- Elementene i feltene kalt Current Temperature og Statistics, samt grafene kalt Temperature History og Histogram er *utgangselementer* eller *indikatorer* (indicators).

Vi ser nå på VI'ens *diagram*, som er vist i figur 2.3 (vi skal ikke gå inn på alle detaljer):

- Den ytre ramme er en *while-løkke* (while-loop). While-løkke er en programstruktur i LabVIEW-språket, som har samme egenskap som while-løkker i andre programspråk, som C, Pascal, Visual Basic m.m. Virkemåten av while-løkke er at programkoden inne i denne ramme (løkke) blir utført om igjen og om igjen så fort LabVIEW makter, helt til løkkes stoppterminal, som er firkanten nederst til venstre i ramme, får logisk tilstand *false*, hvilket skjer når brukeren setter bryteren kalt *Aquisition* på frontpanelet i posisjon *off*.
- Det lille rektangelet markert “TF” er *terminalen* som representerer *Aquisition*-knappen som fins på frontpanelet. “TF” er en forkortelse for “True False”. Terminalens *datatype* er logisk, også kalt boolsk. Når *Aquisition*-knappen står i posisjon *on*, har terminalen verdi *true*. Når *Aquisition*-knappen står i posisjon *off*, har terminalen verdi *false*. Logiske terminaler har grønn farge.

Diagrammet inneholder en rekke *DBL*-terminaler. “DBL” står for “Double”, som uttrykker at datatypen er flyttall med dobbel presisjon. Vi kan derfor omtale disse *DBL*-terminalene som flyttallsterminaler. Også disse terminalene representerer elementer på frontpanelet. Eksempelvis representerer terminalen kalt *High Limit* til venstre i diagrammet skruknappen kalt *High Limit* på frontpanelet. Flyttallsterminaler har oransj farge.

Det fins mange andre datatyper i tillegg til flyttall og logiske data. Disse er omtalt i kap. 5.

- Hvordan kan du se om en terminal i diagrammet representerer et *inngangselement* (*control*) eller en *indikator*? Det er *rammen* rundt terminalen som viser dette:
 - Terminaler for inngangselementer har en tykk strek rundt ramme. Eksempel: Terminalen med navn *High Limit* til venstre i diagrammet.
 - Terminaler for indikatorer har en tynn strek rundt ramme. Eksempel: Terminalen med navn *Mean Temp* til høyre i diagrammet.
- Du kan finne sammenhørende elementer på frontpanel og i diagram slik:
 - Fra frontpanelet: Høyreklikk på elementet / Find Terminal.

- Fra diagrammet: Høyreklikk på terminalen / Find Control eller Find Indicator.
- Inne i while-løkkka, som er en programstruktur, er det ytterligere eksempler på *programstrukturer*:
 - En *for-løkke* (For loop) som sørger for at programkoden inni for-løkkens ramme utføres et bestemt antall ganger, her 10 ganger (siden konstanten 10 er koplet til for-løkkas N-terminal).
 - En *case-struktur* (Case structure) som består av et antall rammer med hvert sitt programinnhold. Verdien av case-strukturens selektor til enhver tid bestemmer hvilken av rammene som skal være aktiv (slik at dens programkode utføres). Case-strukturen svarer til case- og switch-strukturen i andre programspråk.

I tillegg til disse strukturene (while-løkke, for-løkke og case), har LabVIEW en *sekvensstruktur* (sequence).

- Diagrammet inneholder en rekke VI'er, f.eks. **Update Statistics.vi**. Når en VI på denne måten benyttes inne i en annen VI, utgjør førstnevnte en *subVI*.
- Diagrammet inneholder i alt 4 *skiftregistre* (shift registers) som er hukommelselementer som husker verdier av variable (terminaler) slik at disse verdiene blir tilgjengelige i neste utførelse av programkoden i while-løkkka.
- Med hvilken rekkefølge utføres instruksjonene uttrykt i diagrammet? Rekkefølgen er bestemt av *dataflyten* (data flow) som jo skjer langs forbindelseslinjene. En funksjon eller subVI aktiveres eller kalles opp først når alle data som skal inn til funksjonen eller subVI'en, er klare.