

Översikt

I det här projektet ska eleverna bygga och programmera ett första exempel på en återkoppling- och kontrollslina. Denna tekniska princip är central för många industriella system och konsumentprodukter. Återkoppling och kontrollprinciper används också i många av Texas Instruments TI-Innovator™ Hub-projekt. Under projektets gång kommer eleverna att engagera sig i grundläggande koncept inom programmering, såsom variabler, loopar, villkorssatser och booleska operatörer.

Projektet presenteras som en serie utmaningar som bygger de konceptuella kunskaper och färdigheter som behövs för den slutgiltiga öppna utmaningen.

Mål

Eleverna ska

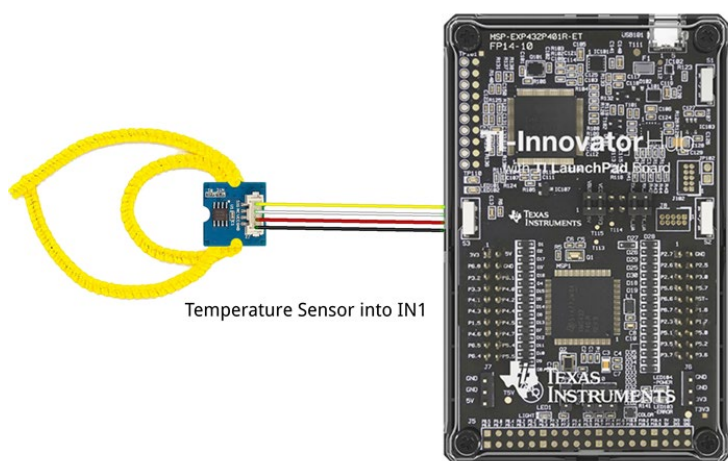
1. Skapa och redigera ett TI-Basic-program på sin handenhet.
2. Koppla ihop hubben med en givare till handenheten
3. Skriva program som innehåller många frekvent använda kommandon
4. Bygga ett enkelt återkopplings- och kontrollsystem

Kommando	Exempel	Förklaring
SET COLOR <r g b>	Send "SET COLOR 255 0 0"	Sätter på lysdioden med röd färg.
DispAt <rad #> , <"text"> , <variabelnamn>	DispAt 3, "Temperatur = ", t	När variabeln t har värdet 26, visas följande rad på räknaren: Temperatur = 26
CONNECT <typ> <tal> TO <port>	Send "CONNECT TEMPERATURE 1 TO IN1"	Associerar det första TEMPERATUR-objektet med en temperaturmodul pluggad i port IN1 på hubben.
READ <typ> <tal>	Send "READ TEMPERATURE 1"	Läser av en mätning från det första temperatur-objektet.
Get <variabel>	Get t	Lagrar temperaturmätningen i en variabel med namnet t. Det lagrade värdet innehåller mätningen från det omedelbart föregående READ kommandot. Obs: Ett Get- kommando måste följa direkt på ett READ-kommando.
For <räkneverksvariabel> , <startvärde> , <slutvärde> , [<stegvärde>] <satser> EndFor	For n,1,10 DispAt 3,n EndFor	Kör en For-loop 10 gånger från 1 till och med 10. Exekverar satsen inom blocket varje gång och visar värdet på räkneverket på rad 3.
While <Booleskt uttryck> <satser> EndWhile	key:=" " While key≠"esc" READ "TEMPERATURE 1" Get t DispAt 3,"Temperature=" ,t key:=getKey() Wait 1 EndWhile	Uttrycken i while-loopen exekveras tills Escapetangenten trycks ned. While-loopen fortsätter så länge det booleska uttrycket utvärderas till "true". Variabeln Key anges till ett initialt värde för en tom sträng med funktionen Assign, ":=". getKey () är en funktion som returnerar en sträng med namnet på den sista nedtryckta tangenten medan ett program körs. I det här programmet är värdet för getKey () lagrat i variabeln key.

<p>If <Booleskt uttryck> Then <satser></p> <p>EndIf</p>	<p>If $t > 26$ Then Send "SET COLOR 255 128 0" EndIf</p>	<p>Kontrollerar för att avgöra om värdet av variabeln t är större än 26. Om uttrycket är sant så exekveras blocket av kommandon i If..</p> <p>EndIf-blocket. Om uttrycket är falskt hoppar blocket över. I det här fallet, när temperaturen är större än 26, kommer räknaren att skicka ett kommando till TI-Innovator för att ställa in färgen på gul.</p>
<p><Boolean expression> and <Boolean expression></p>	<p>If $t \geq 22$ and $t < 24$ Then Send "SET COLOR 0 255 0" EndIf</p>	<p>När båda uttrycken är sanna är "and"-funktionen "true" och satserna exekveras. Annars returnerar funktionen false och satserna hoppas över.</p>

Läs mer på e-guiden för TI-Innovator-teknologi: https://education.ti.com/html/webhelp/EG_Innovator/SV/index.html

Uppställning av försöket



Eleverna kan arbeta i grupper med två eller tre deltagare

Utrustning och materiel

- Räknare
- Enhet till enhet-sladd
- Hubb
- Temperaturgivare
- Grovesladd
- Piprensare

Observera att eleverna behöver trycka piprensaren genom hålen på temperaturgivarplattan för att få en ring. Givarelementet finns på den platta sidan hos givarplattan. Elevdokumentet visar en piprensare men det visas inte så här. Förvänta dig (och uppmuntra) trial and error när eleverna försöker bestämma den mest effektiva ringdesignen.

Elevaktivitet

Sitt i små grupper med din räknare och tillbehören för projektet. Öva på den vägledning som din lärare har gjort.

Lärarkommentarer

Granska och introducera följande räknar- och hubbkommandon som behövs för denna aktivitet med eleverna:

- Starta ett nytt dokument
- Infoga ett program
- Koppla in hubben

Obs: Om eleverna är nybörjare med TI-teknologi och/eller kodning, hänvisa dem då till steg-för-steg-aktiviteter i 10 minutes of Code med TI-Innovator för att komma igång. Det gäller särskilt kapitel 1 Övning 1: Färg och Kapitel 1 Övning 2: Ljud.

<https://education.ti.com/sv/activities/ti-codes/nspire/10-minutes-innovator>

Obs: Videolänken nedan ger en detaljerad steg för steg-demonstration av projektet och utmaningarna (*tal på engelska*):

https://www.youtube.com/playlist?list=PL17Fe0ZmhCR_j09b202PUI0wfaRYSLQka

Om vissa elever är obekanta med stämningsringar eller hur de fungerar, be dem då att göra en snabbsökning på nätet som beskriver hur en stämningsring fungerar.

Du kanske vill börja med denna YouTube-video (*tal på engelska*):

<https://www.youtube.com/watch?v=gvGI51xOKTU&feature=youtu.be>

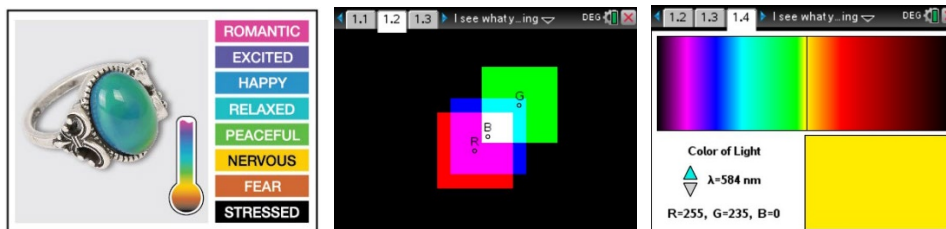
Valfritt: Dela ut den tillhörande elevaktiviteten som innehåller bakgrunds- information och utmaningarna som anges i detta dokument. Genom att göra det kan elevgrupperna arbeta genom utmaningarna i sin egen takt, alternativt kan du välja att visa utmaningarna en i taget på ett annat sätt i ditt klassrum.

Utmaning 1: Använd SET COLOR för att utforska med hjälp av lysdioden Color. Försök att hitta RGB-värdena för alla färger i stämningsring-diagrammet.

t.ex. Send "SET COLOR 255 128 0" gör att det blir gult.

Lärvägledning för Utmaning 1

Demonstrera färgblandningssimuleringarna i TI-Nspire-filen *Color*. Den underlättar förståelsen för additiv färgblandning och RGB-värden. Du kan ladda ner *Color.tns* [härifrån](#).



- Exempelprogram:

```
Define c1a(=
Prgm
Send "SET COLOR 255 128 0"
EndPrgm
```

Utvidgning: Skriv ett program som visar alla färger som stämningsringen ska visa.

```
Define c1b(=
Prgm
Send "SET COLOR 255 0 0"
Wait 2
Send "SET COLOR 255 0 255"
Wait 2
Send "SET COLOR 255 113 0"
Wait 2
Send "SET COLOR 0 0 0"
EndPrgm
```

- *Wait* är användbart för att pausa programmet i ett bestämt antal sekunder
- Försök ställa in färgvärdet till 0 0 0
- Försök ställa in färgvärdet till 225 255 255
- Försök ställa in färgvärdet till 1 1 1

Utmaning 2: Använd kommandot DispAt för att visa ditt namn på flera platser på skärmen.

Lärvägledning Utmaning 2:

- Använd kommandot DispAt (programeditorns I/O-meny). DispAt tar minst två inmatningar (även kallade "argument"). Den första inmatningen är det radnummer där visningen ska ske. Den andra inmatningen är textsträngen, värdet, variabeln etc. som ska visas.

Challenge 2: Use DispAt command to display your name at several locations on the screen.

- Kolon finns på menyn av tecken, tangenten i det nedre högra märkt med "?!".
- Exempelprogram

```
Define c2()=  
Prgm  
DispAt 3,"Hello, my name is:"  
DispAt 4,"Bond"  
EndPrgm
```

Lärvägledning Utmaning 3:

- Loopar används för att upprepa en uppsättning kommandon. En For-loop upprepas ett angivet antal gånger. I TI-Basic definierar programmeraren en For-loop med fyra inmatningar: en räkneverksvariabel, ett startvärde för variabeln, ett slutvärde för variabeln och en valfri variabel för stegvärdet.

- Exempelprogram

```
Define c3()=  
Prgm  
For n,1,10  
DispAt 3,n  
Wait 0.5  
EndFor  
EndPrgm
```

- Stegvärdet sätts till 1 om man inte inkluderar något värde i kommandot.
- Den uppsättning kommandon som ska upprepas definieras i For.. EndFor-blocket.
- For... EndFor finns i programeditorns kontrollmeny (CTL).

Utmaning 4: Ansluta en temperaturgivare till TI-Innovator Hub och visa temperaturen på räknaren

Lärvägledning Utmaning 4:

- Hub Connect-kommandona ger hubben information om vilken typ av enhet som är ansluten till externa portar och hur du kommunicerar med den enheten.
- Kommandot Read instruerar hubben att läsa ett värde från en enhet som är ansluten till en angiven port. Hubben placerar värdet på en minnesplats på hubben och väntar på att räknaren ska "Hämta" värdet.
- Get-kommandot hämtar värdet från hubbens minnesplats för lästa värden och lagrar värdet till en angiven variabel.
- Observera att flera argument kan användas för att skapa ett meddelande med kommandot DispAt.

- Exempelprogram:

```
Define c4()=  
Prgm  
Send "CONNECT TEMPERATURE 1 TO IN 1"  
Send "READ TEMPERATURE 1"  
Get t  
DispAt 4,"Temperatur=",t  
EndPrgm
```

Utmaning 5: Använda en loop för att läsa och visa temperatur.

Lärarvägledning Utmaning 5:

Visa provnumreringen (räkneverksvärdet) utöver temperaturen. Observera att flera inmatningar kan användas för att skapa ett meddelande med kommandot DispAt. Se till att avgränsa argumenten med kommatecken. Obs: Man hittar gradsymbolen i symbolmenyn (tryck Ctrl och boksymbol).

Exempelprogram:

```
Define c5(=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
For n,1,10
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 3,"Prov nr=",n
  DispAt 4,"Temperatur=",t,"°C"
  Wait 2
EndFor
EndPrgm
```

Utmaning 5 Utvidgning:

Använda en loop för att läsa, visa och logga in till list arrays för tid och temperatur. Plotta resultatet.

Lärarvägledning Utmaning 5 Utvidgning:

Obs: Denna utmaning är frivillig. Loggning av givaravläsningar för att lista variabler för senare analys är användbart. Loggning av data krävs dock inte för en fungerande digital stämningssring.

Exempelprogram:

```
Define c5e(=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
time:={}
temp:={}
For n,1,10
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 3,"Tid=",n*0.5,"sek"
  DispAt 4,"Temperatur = ",t,"°C"
  time[n]:=n*0.5
  temp[n]:=t
  Wait 0.5
EndFor
EndPrgm
```

- Avläsningen av temperaturvärdena lagras i en variabel *t*. Sedan under varje For loop-cykel lagras värdet för *t* till position *n* i listvariabeln som heter temp.
- Värdet för tid beräknas genom att multiplicera antalet For loopcykler, *n*, med värdet för Wait, som här är 0,5.
- Variablerna *tid* och *temp* initieras som tomma listor i början av programmet, med hjälp av {}
- Variabeln *n* används för att ange positionen för värdet i listvariablerna, Time [n] och temp [n].
- Värdet för *n* ökas i steg med 1 under varje cykel i While-loopen med denna sats *n*: = *n* + 1. Variabeln *n* anges inledningsvis till 1, den första positionen i TI-Basic-listorna, innan While-loopen börjar.

Utmaning 6: Använd en EndWhile-loop tillsammans med getKey () kommando för att övervaka temperaturen och därefter fatta beslut om att visa meddelandet, "varmt" eller "kallt". Ändra sedan ditt program så att det inkluderar en tredje nivå mellan varmt och kallt, "Skönt".

Lärarvägledning Utmaning 6:

- While-loopar är användbara när du vill ha en uppsättning kommandon som ska utföras när ett visst villkor är sant.
1. While-looptestet kontrollerar om variabeln med namnet Key inte är lika med "esc". Om key inte är lika med "esc" körs kommandona i slingan. Om key är lika med "esc" så avslutar programmet While-loopen och flyttar till nästa kommando efter EndWhile-satsen. I exemplet är variabeln med namnet Key initialt inställd på att vara en tom sträng. Detta försäkrar att While-slingan kommer att köras minst en gång.
 - I While-loopen används en funktion som kallas getKey () för att uppdatera värdet som lagras i variabeln Key. getKey () är en TI-Nspire-funktion som är användbar för tangentbordsstyrda While-Loopar. getKey () lagrar namnet på den sista tangenten som du tryckte på medan ett program körs. Tangentnamnen är värden för textsträngar som omges av citattecken. Till exempel kontrollerar While-looptestet för "esc" inte esc.

Exempelprogram:

```
Define c6a()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:=""
While key≠"esc"
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperatur=",t,"°C"
  If t<22 Then
    Send "SET COLOR 0 0 255"
    DispAt 5,"det är kallt"
  EndIf
  If t≥22 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"det är varmt"
  EndIf
  key:=getKey()
  Wait 0.5
EndWhile
EndPrgm
```

Utvidgning som inkluderar tre temperaturnivåer:

Den booleska operatoren "and" används för att definiera temperaturnivåer. Den och-funktionen kan hittas på katalog-menyn, tangenten med den öppna boken. Se beskrivningen av "and"-funktionen på sid 2 i det här dokumentet. "or"-funktionen är en annan vanligt förekommande boolesk operator. "Or" kan också hittas i katalog-menyn.

```
Define c6b(=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:=""
While key≠"esc"
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperatur=",t,"°C"
  If t<22 Then
    Send "SET COLOR 0 0 255"
    DispAt 5,"det är kallt"
  EndIf
  If t≥22 and t<25 Then
    Send "SET COLOR 0 255 0"
    DispAt 5,"det är skönt"
  EndIf
  If t≥25 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"det är varmt"
  EndIf
  key:=getKey()
  Wait 0.5
EndWhile
EndPrgm
```

Lärarvägledning under den avslutande utmaningen:

- Få eleverna att använda färdigheter från tidigare utmaningar.
- För hur många stämningstillstånd ska elevernas program skrivas?
- Vilka värden ska eleverna använda för att definiera varje stämningssnivå?
- Exempel på grundläggande version av stämningssring-programmet:

```
Define mood(=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:=""
DispAt 8,"Tryck esc för att sluta"
While key≠"esc"
  key:=getKey()
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temp =",t,"°C"
  If t<24 Then
    Send "SET COLOR 0 0 0"
    DispAt 5,"Du är STRESSAD"
```

Slutlig utmaning: Bygg en stämningssring för att upprepade gånger läsa temperaturläsaren, bestämma stämningen hos personen, visa temperaturvärde och visa stämningen.

```
EndIf
  If t≥24 and t<25 Then
    Send "SET COLOR 255 0 0"
  DispAt 5,"Du är NERVOUS"
  EndIf
  If t≥25 and t<26 Then
    Send "SET COLOR 255 150 50"
    DispAt 5,"Du är OSÄKER"
  EndIf
  If t≥26 and t<27 Then
    Send "SET COLOR 0 255 0"
    DispAt 5,"Du är AKTIV"
  EndIf
  If t≥27 and t<28 Then
    Send "SET COLOR 0 255 150"
    DispAt 5,"Du är AVSLAPPNAD"
  EndIf
  If t≥28 and t<29 Then
    Send "SET COLOR 0 255 255"
    DispAt 5,"Du är ÄLSKVÄRD"
  EndIf
  If t≥29 and t<30 Then
    Send "SET COLOR 0 0 255"
    DispAt 5,"Du är ROMANTISK"
  EndIf
  If t≥30 and t<31 Then
    Send "SET COLOR 255 100 75"
    DispAt 5,"Du är GLAD"
  EndIf
  If t≥31 and t<32 Then
    Send "SET COLOR 255 255 0"
    DispAt 5,"Du är TRÖTT"
  EndIf
  If t≥32 Then
    Send "SET COLOR 255 0 255"
    DispAt 5,"Du är LUGN"
  EndIf
  Wait 0.5
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm
```


- Exempel på ett avancerat program, som innehåller många valfria tillägg.
Se förklaringar nedan.

```
Define moodext()=
Prgm
Local a
a:=1
Local t
t:=0
Local room_tempsum
Room_tempsum:=0
Local room_temp
room_temp:=0
Local key
key:=""
Local n
n:=1
time:={ }
temp:={ }
Send "CONNECT TEMPERATURE 1 TO IN 1"
For a,1,8
DispAt a," "
EndFor
DispAt 1,"Stämningssringen"
DispAt 8,"Tryck esc för att sluta"
For a,1,10
Send "READ TEMPERATURE 1"
Get t
room_tempsum:=room_tempsum+t
Wait 0.1
EndFor
room_temp:=room_tempsum/10
DispAt 2,"Rumstemperatur=",round(,1),"°C"
While key≠"esc"
key:=getKey()
Send "READ TEMPERATURE 1"
Get t
DispAt 4,"Temperatur=",round(t,1),"°C"
If t<room_temp+1 Then
Send "SET COLOR 0 0 0"
DispAt 5,"Du är STRESSAD"
Elseif t≥room_temp+1 and t<room_temp+2 Then
Send "SET COLOR 255 0 0"
DispAt 5,"Du är NERVÖS"
Elseif t≥room_temp+2 and t<room_temp+3 Then
Send "SET COLOR 255 150 50"
DispAt 5,"Du är OSÄKER"
```

```

Elseif  $t \geq \text{room\_temp} + 3$  and  $t < \text{room\_temp} + 4$  Then
  Send "SET COLOR 0 255 0"
  DispAt 5,"Du är AKTIV"
Elseif  $t \geq \text{room\_temp} + 4$  and  $t < \text{room\_temp} + 5$  Then
  Send "SET COLOR 0 255 150"
  DispAt 5,"Du är AVSLAPPNAD"
Elseif  $t \geq \text{room\_temp} + 5$  and  $t < \text{room\_temp} + 6$  Then
  Send "SET COLOR 0 255 255"
  DispAt 5,"Du är ÄLSKVÄRD"
Elseif  $t \geq \text{room\_temp} + 6$  and  $t < \text{room\_temp} + 7$  Then
  Send "SET COLOR 0 0 255"
  DispAt 5,"Du är ROMANTISK"
Elseif  $t \geq \text{room\_temp} + 7$  and  $t < \text{room\_temp} + 8$  Then
  Send "SET COLOR 255 100 75"
  DispAt 5,"Du är GLAD"
Elseif  $t \geq \text{room\_temp} + 8$  and  $t < \text{room\_temp} + 9$  Then
  Send "SET COLOR 255 255 0"
  DispAt 5,"Du är TRÖTT"
Elseif  $t \geq \text{room\_temp} + 9$  Then
  Send "SET COLOR 255 0 255"
  DispAt 5,"Du är LUGN"
EndIf
Wait 0.5
 $time[n] := n * 0.5$ 
 $temp[n] := t$ 
 $n := n + 1$ 
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm

```

- Exempel på ett avancerat program, som innehåller många valfria tillägg.
Se förklaringar nedan.

```

Define moodext()=
Prgm
Local  $a := 1$ 
Local  $t := 0$ 
Local  $room\_tempsum : Room\_tempsum := 0$ 
Local  $room\_temp : room\_temp := 0$ 
Local  $key : key := " "$ 
Local  $n := 1$ 
 $time := \{ \}$ 
 $temp := \{ \}$ 
Send "CONNECT TEMPERATURE 1 TO IN 1"
For  $a, 1, 8$ 
  DispAt  $a, " "$ 
EndFor

```

```
DispAt 1,"Stämningssringen"  
DispAt 8,"Tryck esc för att sluta"  
For a,1,10  
Send "READ TEMPERATURE 1"  
Get t  
room_tempsum:=room_tempsum+t  
Wait 0.1  
EndFor  
room_temp:=room_tempsum/10  
DispAt 2,"Room Temperature=",round(,1),"°C"  
While key≠"esc"  
key:=getKey()  
Send "READ TEMPERATURE 1"  
Get t  
DispAt 4,"Temperatur=",round(t,1),"°C"  
If t<room_temp+1 Then  
Send "SET COLOR 0 0 0"  
DispAt 5,"Du är STRESSAD"  
Elseif t≥room_temp+1 and t<room_temp+2 Then  
Send "SET COLOR 255 0 0"  
DispAt 5,"Du är NERVÖS"  
Elseif t≥room_temp+2 and t<room_temp+3 Then  
Send "SET COLOR 255 150 50"  
DispAt 5,"Du är OSÄKER"  
Elseif t≥room_temp+3 and t<room_temp+4 Then  
Send "SET COLOR 0 255 0"  
DispAt 5,"Du är AKTIV"  
Elseif t≥room_temp+4 and t<room_temp+5 Then  
Send "SET COLOR 0 255 150"  
DispAt 5,"Du är AVSLAPPNAD"  
Elseif t≥room_temp+5 and t<room_temp+6 Then  
Send "SET COLOR 0 255 255"  
DispAt 5,"Du är ÄLSKVÄRD"  
Elseif t≥room_temp+6 and t<room_temp+7 Then  
Send "SET COLOR 0 0 255"  
DispAt 5,"Du är ROMANTISK"  
Elseif t≥room_temp+7 and t<room_temp+8 Then  
Send "SET COLOR 255 100 75"  
DispAt 5,"Du är GLAD"  
Elseif t≥room_temp+8 and t<room_temp+9 Then  
Send "SET COLOR 255 255 0"  
DispAt 5,"Du är TRÖTT"  
Elseif t≥room_temp+9 Then  
Send "SET COLOR 255 0 255"  
DispAt 5,"Du är LUGN"  
Endif
```

```
Wait 0.5
time[n]:=n*0.5
temp[n]:=t
n:=n+1
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm
```

Förklaringar för de avancerade programmen

- Variabler kan deklarerars som lokala för programmet. Variabler som deklarerats som lokala kan endast nå inifrån det programmet som de deklarerats i. Globala variabler kan nås från andra program och från alla program i räknaren. Lokala variabler är användbara för att undvika avsiktliga variabelnamnskonflikter och missförstånd. Dessutom så rör lokala variabler inte om i Varsmenyn med extra variabler.
- Flera programkommandon kan placeras på samma rad genom att separera dem med ett kolon. I det här fallet så finns den första uppsättningen variabler på samma rad med ett lokalt kommando och ett startvärde.
- Programmet lagrar temperatur-och tidsvärden i listor för senare analys.
- Programmet använder en For.. EndFor-slinga för att garantera tomma rader för visning av programmets utdata genom att visa tomma textsträngar till var och en av de 8 rader som används av kommandot DispAt.
- Programmet tar hänsyn till temperaturen i omgivningen genom att använda en For.. EndFor-loop för att beräkna medeltemperaturen för 10 avläsningar. Variabeln room_temp används sedan för att skifta temperaturintervallen för stämningsskategorier.
- If-then-Elseif kommandon är användbara när det finns flera nivåer av ömsesidigt uteslutande kategorier för att kontrollera. Kontrollen av Elseif-uttrycken stoppas när en Elseif-Villkorskontroll befinns vara sann. Ett tillvägagångssätt med en serie om.. Sedan.. Endif-satser kommer att fortsätta att kontrollera alla satser i loopen. Den Elseif konstruktion sparar programmet från att göra onödiga kontroller.