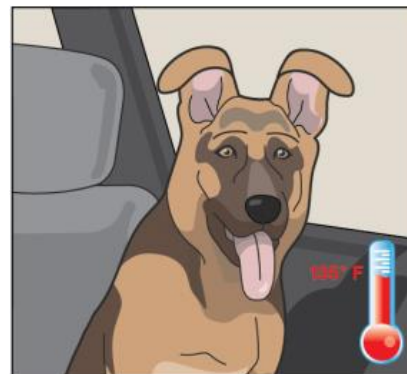


I detta projekt ska eleverna bygga och programmera en enkel återkopplings- och kontrollloop. Återkoppling och kontrollslingar är viktiga för många industriella system och konsumentprodukter. Systemet använder tre ingångsmoduler; två temperatursensorer och en Hall-effekts magnetfältssensor. Systemet har fyra utgångar som inkluderar två vita LED-lampor (strålkastare), 1 kontinuerlig servomotor (kontrollerar fönstret) som även använder sig av TI-Innovator hubbens inbyggda högtalare (horntutande).



Den kontinuerliga servomotorn kräver mer ström än räknaren kan ge. Därför behövs ett externt USB-batteri. Ett program skrivs för att läsa alla tre indataparametrarna och logiskt jämföra dem med kritiska bör-värden för att avgöra när man ska slå på larmet. Projektet presenteras i en rad små utmaningar som är tänkta att stärka kunskapen och färdigheten som krävs för den avslutande öppna utmaningen. Den avslutande utmaningen bygger också på elevernas förståelse i viktiga områden inom biologi och geovetenskap och som är relevanta för att optimera systemet som eleverna slutligen utformar och förfinar.

Obs: detta projekt förutsätter att eleverna har praktisk erfarenhet av viktiga programmeringsbegrepp i *Digital Mood Ring projektet*. Referera gärna till detta projekt. Dokumentation för Digital Mood Ring [finns här](#)

Tabellen nedan visar den uppskattade temperaturen inne i bilen vid olika utomhustemperaturer och efter olika tider med stängd bil.

	A	B	C	D	E	F	G
1	tid/utomhustemp	21.	24.	27.	29.	32.	35.
2	0 min	21.	24.	27.	29.	32.	35.
3	10 min	32.	34.	37.	40.	43.	46.
4	20 min	37.	40.	43.	46.	48.	51.
5	30 min	40.	43.	46.	48.	51.	54.
6	40 min	42.	45.	48.	51.	53.	56.
7	50 min	44.	47.	49.	52.	55.	58.
8	60 min	45.	48.	51.	53.	56.	59.
9	> 60 minuter	46.	49.	52.	54.	57.	60.
10							
11							

Ljus till värme

Varje sommar läser vi fruktansvärda historier om barn och husdjur som lämnas kvar i varma bilar. Det slutar oftast med värmeslag, men i sällsynta fall slutar det i dödsfall. Insidan av en bil värms upp mycket snabbare än utsidan, på grund av växthuseffekten. Solstrålningen strömmar in i fordonet genom fönstren och träffar ytorna på insidan av bilen. Synligt ljus absorberas och strålar tillbaka som infrarött ljus. Infraröd strålning har en större våglängd än synligt ljus och den strålningen är oförmögen att stråla tillbaka genom fönstren. Den infångade strålningen gör att temperaturen inuti bilen då stiger snabbare än utomhustemperaturen.

Staying cool!

Däggdjur som människor, hundar och katter har alla sätt att reglera temperaturen, termoregulering. Kroppen har förmåga att hålla sig stabil. Det kallas för *homeostas*. Människor svettas för att öka avlägsnandet av värme från kroppen genom avdunstning. Hundar brukar flämta för att ta bort värme även om de också har ett litet antal svettkörtlar i kuddar hos sina tassar. Katter lägger sig på ytor som är relativt kalla för att hjälpa till att ta bort kroppsvärmen. De slickar också sina tassar och gnider saliven på varmare delar för att öka avdunstningen. Det är en liknande mekanism som svettning hos människor. När de termiska regleringsmekanismerna inte kan upprätthålla homeostas kommer vi däggdjur att gå in i tillstånd som kallas värmeslag.

Problemet med för mycket värme

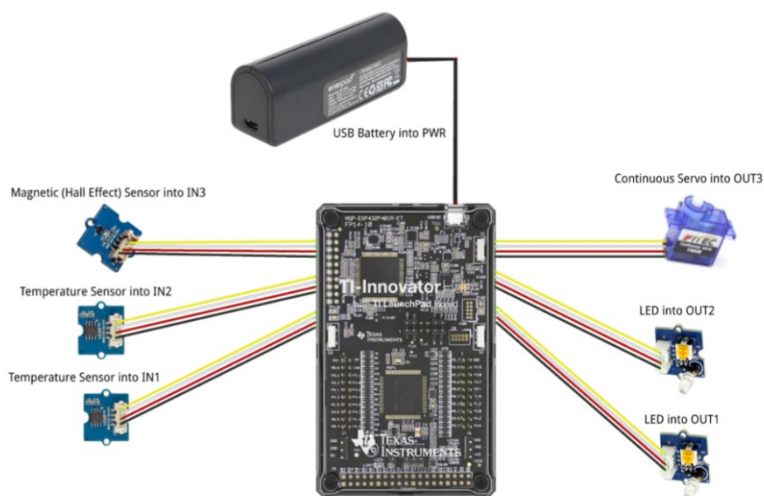
Värmeslag leder till funktionsnedsättning i hjärnan, uttorkning, hjärtsvikt, cellsvullnad och eventuellt död.

Låt eleverna i grupp bedriva lite forskning om varför däggdjur som människor, hundar och katter inte kan uthärda varma miljöer som slutna bilar på sommaren under långa perioder. De ska sedan utforma en lösning med hjälp av teknologi för att skydda bilpassagerare från värmeslag. Be dem förfina sin design tills de har en fungerande prototyp. Med lärares tillåtelse ska de sedan jämföras sin prototyp med övriga klasskamrater för att avgöra vilket lag som har det "bästa" systemet.

Kommando	Exempel	Beskrivning
CONNECT <typ> <tal> TO <port>	Send "CONNECT TEMPERATURE 1 TO IN1"	Associerar det första TEMPERATUR-objektet med en temperaturmodul ansluten till port IN1 hos hubben.
SET <typ><tal> TO <värde>.	Send "SET LED 1 TO ON"	Sätter på LED 1. Andra parametrar 1 kan inkludera BLINK <hastighet> TIME <varaktighet> "SET LED 1 ON BLINK 3 TIME 20"
SET <typ><tal> TO <värde> [<blinkningshastighet >] [<varaktighet i sekunder>]	Send "SET LED 1 ON BLINK 3 TIME 20"	Sätter LED 1 på en blinkningshastighet på 3 gånger per sekund i 20 sekunder. Se menyn för hubbinställningar för ON, OFF, BLINK, etc.
SET <typ> <ta> TO <värde>	Send "SET SERVO.CONTINUOUS 1 CW 20 TIME 1"	Sätter på det första kontinuerligt roterande servomotor-objektet vid effekten 20 (område 0-100) i medurs riktning (CW (clockwise) vs. CCW (counterclockwise) i 1 sekund.
Wait <number>	Wait 3	Pausar programmet ett bestämt antal sekunder. I detta fall 3 sekunder.
READ <typ> <tal>	Send "READ TEMPERATURE 1"	Läser av en mätning från det första temperaturobjektet.
Get <variable>	Get <i>t</i>	Lagrar temperaturmätningen i en variabel som heter <i>t</i> . Värdet som lagras kommer att innehålla mätningen från det närmast föregående READ-kommandot. Obs: ett Get-kommando måste omedelbart följa på ett READ-kommando.
DispAt <rad #> , <"text"> , <variabelnamn>	DispAt 3, "Temperatur = ", <i>t</i>	När variabeln <i>t</i> har värdet 26 visas följande rad på räknaren: temperatur = 26 (temperaturavläsningar är i °C som standard)
For <räkneverksvariabel> , <startvärde> , <slutvärde> , [<steg>] <satser> EndFor	For <i>n</i> , 1, 10 DispAt 3, <i>n</i> EndFor	Körs For-loopen 10 gånger, börjar vid 1 och slutar vid 10. Exekverar uttrycket i blocket varje gång, visar värdet för räkneverksvariabeln på rad 3.
While <Booleskt uttryck> <satser> EndWhile	key:=" " While key≠"esc" READ "TEMPERATURE 1" Get <i>t</i> DispAt 3, "Temperature= ", <i>t</i> key:=getKey() Wait 1 EndWhile	Uttrycken i While-loopen exekveras tills Escape-tangenten trycks ned. While-loopen fortsätter så länge det booleska uttrycket utvärderas till "true". Variabeln <i>key</i> anges till ett initialt värde för en tom sträng med tilldelningsfunktionen "=". getKey () är en funktion som returnerar en sträng med namnet på den sista tangenten som trycktes ned medan ett program körs. I det här programmet är värdet för getKey () lagrat i variabel <i>key</i> .
<Booleskt uttryck> and <Booleskt uttryck>	If <i>t</i> >25 and <i>m</i> <100 Then Send "SET COLOR 0 255 0" EndIf	När båda uttrycken är sanna är "and"-funktionen "true" och satserna exekveras. Annars returnerar funktionen false och satserna överhoppas.
If <Booleskt uttryck> Then <sats 1> Else <sats 2> End	If <i>m</i> <100 Then DispAt 5, "Magnet finns" Else DispAt 5, "Magnet finns inte" EndIf	Exemplet med beslutsträdet har ett booleskt uttryck med motsvarande satser att köra om sant. Det har också ett Else-villkor som kör motsvarande satser när det booleska uttrycket är falskt. Den här Else-villkoret säkerställer att en uppsättning satser alltid kommer att utföras. När det här beslutsträdet körs går fokus i riktning uppifrån och ned. Om värdet på <i>m</i> är mindre än 100 exekveras kommandona efter Then. I det här fallet visas utmatningen "Magnet finns" på rad 5. Om värdet för <i>m</i> är större än eller lika med 100 exekveras kommandona efter Else. I detta fall visas utmatningen "Magnet finns inte" på rad 5.

1. Räknare

2. Enhet till enhetskabel
3. TI-Innovator Hub
4. Grove kabel x5
5. Grove temperaturgivare x2
6. Dockbil, skolåda, eller ett annat objekt för att modellera en bil
7. Grove magnetisk närhetssensor (Halleffekt)
8. Grove vitt LED-ljus x2
9. Grove kontinuerligt roterande Servomotor
10. Externt USB-batteri w/kabel



Utmaning 1: Skriv ett program som spelar två ljud i 1 sekund vardera i en For-loop som upprepas fem gånger.

T.ex. Send "SET SOUND 220"

Elevaktivitet

Lärarkommentarer

Läroanvisning för Utmaning 1:

- Använd en For.. EndFor slinga att växla mellan två olika ljudfrekvenser som är en oktav isär. Loopar används för att upprepa en kommandouppsättning. En For-loop upprepar ett angivet antal gånger. I TI-Basic definieras en For-loop med fyra inmatningar: en räkneverksvariabel, ett startvärde för variabeln, ett slutvärde för variabeln och en valfri variabel för stegvärdet.
- For... EndFor finns i programmets kontrollmeny
- Använd Wait-kommandot för att bestämma hur länge varje ton ska spelas
- Exempel på ett program:

```
Define pc1()=
Prgm
  For n,1,10
    Send "SET SOUND 440"
    Wait 1
    Send "SET SOUND 880"
    Wait 1
  EndFor
EndPrgm
```

- Försök nu att lägga till en tredje ton i programmet
- Försök att hitta gränserna för den inbyggda högtalaren genom att ändra frekvenserna

Utmaning 2a: Använd For..EndFor för att få två externa LED att blinka.

Läroanvisning för Utmaning 2:

- Hub Connect-kommandona ger hubben information om vilken typ av enhet som är ansluten till externa portar och hur du kommunicerar med den enheten.
- Anslut lysdioder till OUT 1 och OUT2 på hubben
- Använd kommandot Send "CONNECT LED # TO OUT #"
- Använd ett For.. EndFor-kommando för att få lysdiодerna att slå av och på
- Exempel på ett program:

```
Define pc2a()=  
Prgm  
Send "CONNECT LED 1 TO OUT 1 "  
Send "CONNECT LED 2 TO OUT 2 "  
For n,1,30  
  Send "SET LED 1 ON"  
  Send "SET LED 2 ON"  
  Wait 1  
  Send "SET LED 1 OFF"  
  Send "SET LED 2 OFF"  
  Wait 1  
EndFor  
EndPrgm
```

Utmaning 2b: Försök nu med Utmaning 2a en gång till och använd nu "BLINK" och "TIME". Denna gång ska du inte använda en For-loop.

- Exempel på ett program:

```
Define pc2b()=  
Prgm  
Send "CONNECT LED 1 TO OUT 1 "  
Send "CONNECT LED 2 TO OUT 2 "  
Send "SET LED 1 ON BLINK 3 TIME 30"  
Send "SET LED 2 ON BLINK 3 TIME 30"  
Wait 30  
EndPrgm
```

Utmaning 3: Anslut en kontinuerligt roterande servomotor till hubben och få den att rotera medurs (CW) och sedan i motsatt riktning, moturs (CCW).

Läroanvisning för Utmaning 3:

- Använd en for.. EndFor loop för att få en kontinuerligt roterande servomotor att rotera medurs och sedan moturs.
- Använd Send "CONNECT SERVO. CONTINUOUS # TILL OUT 3 "
- SERVO CONTINUOUS-motorn måste anslutas till OUT 3-porten eftersom 5 volt är nödvändigt för att driva servomotorn.

- Använd medurs (CW) eller moturs (CCW) för att ställa in sväng-riktningen för den kontinuerligt roterande servomotorn.
- Kraften för den kontinuerligt roterande servomotorn varierar från 0 till 100. 0 är avstängd och 100 är full effekt.
- Se till att ett externt batteri är anslutet till PWR-porten på hubben och att det är aktiverat precis innan programmet körs.
- Exempel på ett program

```
Define c3()=  
Prgm  
Send "CONNECT SERVO.CONTINUOUS 1 TO OUT 3"  
For n,1,5  
  Send "SET SERVO.CONTINUOUS 1 CW 20 TIME 1"  
  Wait 2  
  Send "SET SERVO.CONTINUOUS 1 CCW 20 TIME 1"  
  Wait 2  
EndFor  
EndPrgm
```

- Experimentera med att ändra effekt och värdet på TIME och tillhörande väntetid för att se vilken effekt det har på hur många varv servomotorn kommer att göra

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

Läroanvisning 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.
- Medan loopar är användbara när du vill ha en uppsättning kommandon som ska utföras när ett visst villkor är sant.
 - 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" 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-slingan 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 programmet kördes. Key-namnen är textsträng-värden som omges av citattecken. Till exempel kontrollerar while-looptestet för "esc" inte esc.
- Observera att flera argument kan användas för att skapa ett meddelande med kommandot DispAt.
 - Exempel på ett program:

```
Define c4()=  
Prgm  
Send "CONNECT TEMPERATURE 1 TO IN 1"  
DispAt 8,"Tryck [esc] för att avsluta"  
key:=""  
  
While key≠"esc"  
Send "READ TEMPERATURE 1"  
Get t  
DispAt 3,"Temperatur=",t,"°C"  
Wait 0.5  
key:=getKey()  
EndWhile  
EndPrgm
```

Utmaning 5: Anslut den magnetiska närhetssensorn (Halleffekt), som avgör om sydpolen hos ett magnetfält är nära sensorn. Visa "Magnet finns " eller "Magnet finns inte " baserat på avläsning av Halleffekt-sensorn och positionen hos magneten.

Läroanvisning Utmaning 5:

- Vissa sensorer eller utdataenheter har inte ett associerat objekt som finns i input-och output-menyer. För dessa enheter använder vi ANALOG.IN, DIGITAL.IN, ANALOG.OUT eller DIGITAL.OUT.
- Halleffektsensorn detekterar närvaron (eller frånvaron) av ett magnetfält med sydlig polaritet. För Hallsensorn använder du ANALOG.IN.
- Sensorvärdena är relaterade till spänningsmätningar. När Sydpolen på magneten är nära ligger avläsningarna oftast under 100.
- Det rekommenderas att IN3 används för att förse sensorn tillräckligt med ström.
- Anmärkningar: ©-symbolen betecknar en kommentar. Kommentarer körs inte när programmet körs. Från Programeditorn väljer du *Åtgärder* och sedan *Infoga kommentar*.

- Exempel program:

```
Define c5()=
Prgm
Send "CONNECT ANALOG.IN 1 TO IN 3"
© ANALOG.IN 1 är objektet för den magnetiska närhetssensorn
DispAt 8,"Tryck [esc] för att avsluta"
key:=" "
While key≠"esc"
Send "READ ANALOG.IN 1"
Get m
DispAt 3,"Magnetic sensor(bits)= ",m
If m<100 Then
DispAt 5,"Magnet finns"
Else
DispAt 5,"Magnet finns inte"
EndIf
Wait 0.5
key:=getKey()
EndWhile
EndPrgm
```

Avslutande Utmaning: Använd de färdigheter som utvecklats i de tidigare utmaningarna för att utveckla ett larmsystem för en modell av en bil som avgör om ett husdjur är närvarande (magnet) OCH temperaturavläsning inuti bilen när temperaturen når en kritisk tröskel innan larmet utlöses.

Läroanvisning Avslutande Utmaning:

- I denna utmaning ska eleverna skapa ett system som läser data från ingångar såsom temperatur och Hallsensorn som i sin tur styr lysdioder, ljud genom högtalaren, och den kontinuerligt roterande servomotorn.
- Observera att While.. EndWhile används ofta med sensorer (ingångar) och If.. EndIf används med utgångar.
- Exempel på program:

```
Define c6()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN1"
Send "CONNECT TEMPERATURE 2 TO IN2"
Send "CONNECT ANALOG.IN 1 TO IN3"
Send "CONNECT LED 1 TO OUT1"
Send "CONNECT LED 2 TO OUT2"
Send "CONNECT SERVO.CONTINUOUS 1 TO OUT3"
DispAt 1,"Alarmet är ON"
DispAt 8,"tryck på esc för att avsluta övervakningen av bilen"
key:=" "
window:=1
While key≠"esc"
Send "READ TEMPERATURE 1"
Get temp_in
Send "READ TEMPERATURE 2"
```



```
Get temp_out
Send "READ ANALOG.IN 1"
Get m
If m<100 Then
  DispAt 3,"Husdjur I bilen"
Else
  DispAt 3,"Inget husdjur I bilen"
EndIf
If window=1 Then
  DispAt 4,"Fönstret är stängt"
Else
  DispAt 4,"Fönstret är öppet"
EndIf
DispAt 5,"Temperatur inne I bilen= ",temp_in,"°C"
DispAt 6,"Temperatur utanför bilen= ",temp_out,"°C"
If temp_in>25 and m<100 Then
  DispAt 2 "Varning! Husdjur har fått värmeslag!"
  Send "SET LED 1 ON BLINK 3"
  Send "SET LED 2 ON BLINK 3"
  Send "SET SOUND 440 TIME .5"
  Wait 0.5
  Send "SET SOUND 880 TIME .5"
  Wait 0.5
  If window=1 Then
    Send "SET SERVO.CONTINUOUS 1 CW 30 .25"
    Wait .25
    window:=0
  EndIf
Else
  DispAt 2 "Ingen varning"
  Send "SET LED 1 OFF"
  Send "SET LED 2 OFF"
  Send "SET SOUND OFF"
  If window=0 Then
    Send "SET SERVO.CONTINUOUS 1 CCW 30 .25"
    Wait .25
    window:=1
  EndIf
EndIf
EndIf
key:=getKey()
EndWhile
Send "SET LED 1 OFF"
Send "SET LED 2 OFF"
Send "SET SOUND OFF"
Send "SET SERVO.CONTINUOUS 1 0"
DispAt 1,"Alarmet är i läge OFF"
DispAt 2," "
EndPrgm
```