

Smart bevattningssystem

I detta TI-Innovator™ projekt, ska du utforma ett smart bevattningssystem som kan användas för att övervaka och uppmäta vatten från en regnuppsamlingscistern som kan användas för att bevattna en liten familjeträdgård i Zimbabwe. Den här modellen gäller även andra relaterade scenarier där "smartare vatten" kan användas. Till exempel förekommer det ofta vattenrestriktioner under de varma sommarmånaderna i områden med mycket varmt och torrt klimat. Ett smart vattenbevattningssystem bör kunna lätta på vissa av dessa restriktioner.

Du kommer att behöva använda dina matematiska färdigheter, datorprogrammering och allmän ingenjörskonst för att designa och bygga ett smart bevattningssystem för att lösa de problem som torkan orsakar i Zimbabwe.

Bakgrund:

En torka i södra Afrika har orsakat hungersnöd i Zimbabwe och många skolungdomar tvingas sluta skolan och stanna hemma för att hjälpa till med odlingen av matgrödor för familjen. Vad kan göras annorlunda när det gäller bevattning av grödor så att skoleleverna kan vara i skolan i stället för att tvingas arbeta på fälten?

Din utmaning:

Blir att hjälpa till att lösa detta problem genom att designa och bygga ett smart bevattningssystem för att hantera en begränsad mängd vatten som samlats i en cistern för att bevattna en trädgård.

Material

- TI-Innovator Hub och kabel
- TI Nspire CX handenheter
- Grove - Temperatur & fuktighets sensor
- Grove – Jordfuktsensor
- Grove –Ljussensor
- Grove– MOSFET för strömkontrollmodul med 4xAA batterihållare
- 4 x AA batterier för att förse ström till pumpen (behövs)
- Vattenpump med plaströr
- Möjliga tillbehör för att bygga trädgårdsmodell:
 - Sugrör
 - Silvertejp (alltid användbart)
 - Behållare för växterna
 - Jord, perlit eller någon annan odlingsbas

Projekt utmaningar

Utmaning 1: Skriv ett program som heter C1 som kontinuerligt mäter och visar den omgivande ljusnivån. Programmet ska:

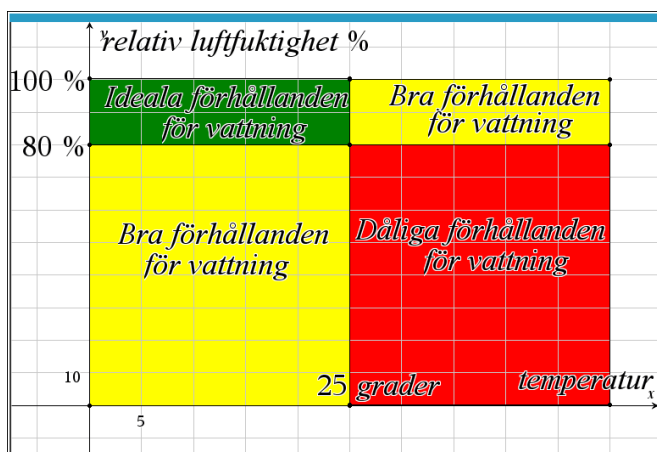
1. Ansluta en ljus nivågivare till IN 1.
2. Skala ljusnivån för avläsning från 0 till 100.
3. Använd en while-loop för att kontinuerligt läsa av och visa ljusnivån varje halvsekund på räknarens skärm.
4. Aktivera Esc-tangenten för att avsluta while-loopen och avsluta programmet.



Utmaning 2: Skriv ett program som heter C2 som mäter jordfuktighet varannan sekund, sammanlagt tjugo gånger. Programmet ska visa om jorden är torr eller fuktig baserat på sensorns avläsning.

Utmaning 3: Skriv ett program som heter C3 som förbinder temperatur- och luftfuktighetssensorn (DHT). Gör 20 mätningar med två sekunders mellanrum och visa med ett lämpligt meddelande.

1. Använd ett beslutsträd baserat på mätningar av temperatur och relativ luftfuktighet för att bestämma de aktuella vattningsförhållandena, som anges i grafen nedan. Visa ett lämpligt meddelande för vart och ett av de fyra fallen.



Utmaning 4: Skriv ett program som heter C4 och som använder en while-loop för att kontinuerligt mäta och visa en instrumentpanel av alla avläsningar av sensorvärden. Användaren ska kunna stoppa övervakningen genom att trycka på Esc-tangenten.

Utmaning 5: Skriv ett program som heter C5 för att ansluta pumpens strömmodul och köra pumpen i 20 sekunder. Var noga med att stänga av pumpen.

1. Försök att ställa in pumpens effekt på olika värden.
2. Försök att uppskatta pumpens flödes hastighet i ml/s.

Avslutande utmaning: Skriv ett program som använder alla sensorer från de tidigare utmaningarna. Övervaka kontinuerligt ljusnivå, markens fuktighet, temperatur och luftfuktighet och visa det aktuella värdet på displayen. Använd dina kunskaper om ekologi, biologi och geovetenskap för att avgöra vilka de bästa förutsättningarna är för att vattna din trädgård. När villkoret är korrekt så ställ in pumpen så att den levererar vatten i en takt som är optimal för din trädgård.

Testa ditt system med dessa villkor:

- ljusnivå < 20
- jordfuktighet < 10
- temperatur < 25
- luftfuktighet > 80

När dessa villkor är sanna, ställ in pumpen på 255. När villkoren är falska, ställ in pumpen på noll.

Exempel	Beskrivning
Send "CONNECT LIGHTLEVEL 1 TO IN1"	Associerar det första LIGHTLEVEL-objektet med en ljussensor ansluten till port IN1 hos hubben.
Send "SET ANALOG.OUT 1 TO 128"	Sätter på analog out1-objekt, som till exempel en pump till en strömkontrollinställning på 128.
Send "RANGE LIGHTLEVEL 1 0 100"	Skalar de lästa uppmätta värdena från LIGHTLEVEL 1 returnerade i intervallet 0 till 100
Send "READ MOISTURE 1"	Läser av en mätning från den första fuktsensorn.
Get m	Lagrar fuktmätningen i en variabel som heter m. Obs: Ett Get-kommando måste omedelbart följa på ett READ-kommando. Värdet som lagras kommer att innehålla mätningen från det närmast föregående READ-kommandot.
DispAt 3, "Markfuktnivå = ", m	När variabeln m har värdet 26 visas följande rad på rad 3 på räknaren: Markfuktnivå = 26.
key:=" " While key≠"esc" SEND "READ MOISTURE 1" Get m DispAt 3, "Markfuktnivå= ", m key:=getKey() EndWhile	Kommandona i While-loopen loopas tills Escape-tangenten trycks ned. While-loopen fortsätter så länge det logiska uttrycket $key \neq "esc"$ är sant. Variabeln key är initialt angiven som ett tomt värde så while-loopen kommer att exekveras minst en gång. getKey()-funktionen övervakar tangentbordet och returnerar en sträng med samma namn som den sist nedtryckta tangenten.
If t>=40 Then DispAt , "Det är Varmt" Elseif t>=25 and t<40 Then DispAt 3, "Det är Varmt" Else DispAt 3 "Det är Kallt" EndIf	Exemplet med beslutsträdet har två ömsesidigt uteslutande booleska uttryck och två motsvarande exekveringsatser. Det har också ett Else-villkor som utför motsvarande satser när inget av de två första villkoren är sanna. Detta sista Else-villkor säkerställer att en uppsättning satser alltid kommer att utföras. När det här beslutsträdet körs fortsätter fokus uppifrån och ned. Om det första booleska uttrycket är sant exekveras motsvarande satser och beslutsträdet avslutas omedelbart. I exemplet: om $t = 30$ är det första uttrycket falskt och <satser 1> hoppas över, det andra uttrycket är sant och <satser 2> körs och trädet avslutas. Ytterligare Elseif-satser får vid behov införas.
If luftfuktighet>=50 and ljus<10 Then DispAt 3, "Pumpen är På" EndIf	"and"-operatorn jämför två separata Booleska uttryck samtidigt. När båda uttrycken är sanna är operationen sann.
If light<50 or markfukt<10 Then DispAt 3, "Pumpen är på" EndIf	"Or"-operatorn jämför två separata Booleska uttryck samtidigt. När något eller båda uttrycken är sanna är operationen sann.

Kalkylatorkommandon:

- På startskärmen trycker du på **4: Aktuellt** för att återgå till dokumentfilen.
- På startskärmen trycker du på **1: Nytt** för att skapa en ny dokumentfil.
- Du skapar och redigerar program i en Progradeditor-app. Du kör sedan programmet inifrån en Räkna-app.
- Använd [**menu**]-knappen för att se alternativen för din nuvarande app.
- **CTRL-b** är genvägen för Kontrollera syntax och lagrar menyn för att lagra ändringar i ditt program.
- **CTRL-r** är genvägen från Kontrollera syntax och lagra-menyn för att lagra ändringar i ditt program och klistra in namnet i en Räkna-app
- Tryck på [**enter**] för att köra ett program som inmatats på Räkna-appens inmatningsrad.
- Appen Räkna "kommer ihåg" det sista kommandot som angavs. Tryck på [**enter**] när ett program har körts och du vill köra programmet igen.
- Hitta dina programnamn i Räkna-appen genom att trycka på [**var**] (står för variabler).
- Flytta från sida till sida genom att använda [**ctrl**]-vänsterpil och [**ctrl**]-högerpil eller genom att använda pekplattpekaren för att klicka på önskad sidflik.
- [**ctrl**]-[**doc**] (+ Page) kommer att lägga till en tom sida i ditt dokument.
- [**ctrl**]-z kommer att ångra din senaste åtgärd.
- För att stoppa ("Break") ett program tryck och håll på knappen [**ON**] tills du får en dialogruta.
- [**ctrl**]-s är genvägen för att spara hela dokumentfilen. Gör detta med jämna mellanrum för att spara ditt arbete.

Sensorer och ställdon Hubb-anslutningar

