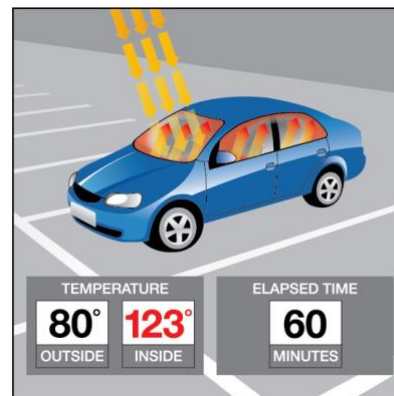


## Pet Car Alarm

In this TI-Innovator™ project, you will explore the science behind the greenhouse effect and apply your knowledge to design a product to solve a real world problem of pets dying due to owners leaving them in hot cars. You will have to utilize math skills, computer programming and engineering to design and build a smart pet alarm system for a model car. A car equipped with a “smart” pet alarm could prevent harm to a pet left inside a hot car by taking action to cool the interior and notify the owner of impending pet harm.



### Background:

Pets suffer when left unattended in a car with the windows rolled up on a hot sunny day. The temperatures inside a car may reach greater than 40°F above the outside ambient temperature within an hour due to a greenhouse effect within the closed car. A car equipped with a pet-smart alarm could prevent harm to a pet left inside a hot car by taking action to cool the interior and notify the owner of impending pet harm.

### Your Challenge:

Understand the science behind the greenhouse effect and use math, computer programming and engineering to design and build a pet-smart alarm system using a model car.

### Activity Materials:

- TI-Innovator™ Hub (vs.1.3)
- TI-Nspire CX (OS vs. 4.5)
- Hall Effect (magnetic) Sensor
- Temperature Sensor(s)
- 2 White LEDs
- Continuous Servo Motor
- External Battery for TI-Innovator Hub
- Small magnet
- Model Car (or other container)
- Plastic Pet
- Cling wrap
- Tape
- Safety Scissors

## Project Challenges

**Skill Building Challenge 1:** Write a program that will play two sounds for 1 second each in a For loop that repeats five times. e.g. Send “SET SOUND 220”

**Skill Building Challenge 2a:** Use a For..EndFor loop to blink two external LED's.

**Skill Building. Challenge 2b:** Try Challenge 2a once more using the “BLINK” and “TIME” options. This time you can do it without a For Loop.

**Skill Building Challenge 3:** Connect a continuous servo motor to the TI-Innovator Hub and cause it to rotate clockwise (CW) and then in the opposite direction, counterclockwise (CCW).

**Skill Building Challenge 4:** Connect a temperature sensor to the TI-Innovator Hub and display the temperature on the calculator.

**Skill Building Challenge 5:** Connect the Hall effect magnetic proximity sensor, which determines if the south pole of a magnetic field is close to the sensor. Display “Magnet is present” or “Magnet is not present” based on the reading of the Hall effect sensor and the position of the magnet.

**Final Challenge:** Use the skills developed in the earlier challenges to develop an alarm system for a model of a car that determines if a pet is present (magnet) AND the temperature reading inside the car reaches a critical threshold before triggering the alarm.

**Example Programming Commands for the project**

Example Code	Behavior
Send "CONNECT TEMPERATURE 1 TO IN1"	Associates the first TEMPERATURE object with a temperature module plugged into port IN1 on the Hub.
Send "SET LED 1 TO ON"	Turns on LED 1. Other parameters may include BLINK <rate> TIME<duration> "SET LED 1 ON BLINK 3 TIME 20"
Send "SET LED 1 ON BLINK 3 TIME 20"	Sets LED 1 to a blink rate of 3 times per second for 20 seconds. See Hub Settings menu for ON, OFF, BLINK, etc.
Send "SET SERVO.CONTINUOUS 1 CW 20 TIME 1"	Turns on the first continuous servo motor object at power 20 (range 0-100) in the clockwise direction (CW vs. CCW) for a time of 1 second.
<pre> Wait 3  Send "READ TEMPERATURE 1"  Get t  DispAt 3, "Temperature = ", t           </pre>	<p>Pauses the program for a specified number of seconds. In this case 3 seconds.</p> <p>Reads a measurement from the first temperature object.</p> <p>Stores the temperature measurement into a variable named <i>t</i>. The value stored will contain the measurement from the immediately preceding READ command. Note: a Get command must immediately follow a READ command.</p> <p>When variable <i>t</i> has a value of 26, the following line is displayed on the calculator: Temperature = 26 (temp readings are in °C by default)</p>
<pre> For n,1,10   DispAt 3,n EndFor  key:="" While key!="esc"   READ "TEMPERATURE 1"   Get t   DispAt 3,"Temperature= ",t   key:=getKey()   Wait 1 EndWhile  If t&gt;25 and m&lt;200 Then   Send "SET COLOR 0 255 0" EndIf           </pre>	<p>Runs For loop 10 times, starting at 1 and ending at 10. Executes the statement in the block each time, displays the value for the counter variable on row 3.</p> <p>The statements in the While loop are executed until the escape key is pressed. The While loop continues as long as the Boolean expression evaluates to "true". The variable <i>key</i> is set to an initial value of an empty string using the assign function, ":=". getKey() is a function that returns a string with the name of the last key pressed while a programming is running. In this program the value of getKey() is stored to the variable <i>key</i>.</p> <p>When both expressions are true the "and" function is "true" and the statements are executed. Otherwise, the function returns false and the statements are skipped.</p>

```
If m<200 Then
DispAt 5,"Magnet is present"
Else
DispAt 5,"Magnet is not present"
EndIf
```

The example decision tree has a Boolean expression with corresponding statements to execute if true. It also has an Else condition that executes corresponding statements when the Boolean expression is false. This Else condition ensures that a set of statements will always be executed. When this decision tree executes, focus proceeds from top-down. If the value of  $m$  is less than 200 the commands after Then are executed. In this case the output "Magnet is present" is displayed on row 5. If the value of  $m$  is greater than or equal to 200 the commands after Else are executed. In this case the output "Magnet is not present" is displayed on row 5.

**Calculator Notes:**

- On the Home screen press 4:Current to return to your document file.
- On the Home screen Press 1:New to create a new document file.
- You create and edit programs in a Program Editor app. You run programs from within a Calculator app.
- Use the [menu] key to see the options for your current app.
- ctrl-b is the shortcut from the Check Syntax and Store menu to store changes to your program.
- ctrl-r is the shortcut from the Check Syntax and Store menu to store changes to your program and paste the name to a Calculator app
- Press [enter] to run a program named on a Calculator app entry line.
- The Calculator app "remembers" the last command entered. Press Enter after a program has run to run the program again.
- Find your program names in a Calculator app by pressing the [var] (variables) key.
- Move from page to page by using ctrl-left arrow and ctrl-right arrow or by using the touchpad pointer to click on the desired page tab.
- ctrl-doc (+page) will add a blank page to your document.
- ctrl-z will undo your last action.
- To stop ("break") a program press and hold the ON key until you receive a dialogue box.
- ctrl-s is the shortcut for saving your entire document file. Do this periodically to save your work.

**Sensor and actuator Hub connections**

