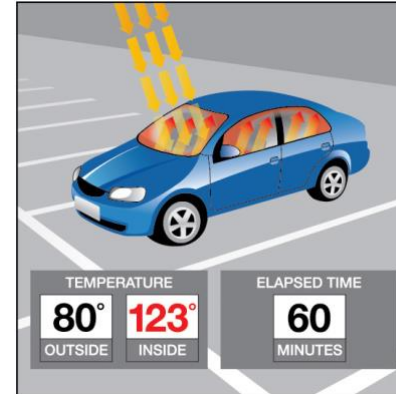


Pet Car Alarm

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-84 Plus CE (OS vs. 5.3)
- 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 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 |
|---|---|
| <code>Send("CONNECT TEMPERATURE 1 TO IN1")</code> | Associates the first TEMPERATURE object with a temperature sensor plugged into port IN1 on the Hub. |
| <code>Send("SET LED 1 ON")</code> | Sets LED 1 on. See Hub Settings menu for ON, OFF, BLINK, etc. |
| <code>Send("SET LED 1 ON BLINK 3 TIME 20")</code> | Sets LED 1 to a blink rate of 3 times per second for 20 seconds. See Hub Settings menu for ON, OFF, BLINK, etc. |
| <code>Send("SET SERVO.CONTINUOUS 1 CW 20 TIME 1")</code> | 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. |
| <code>Wait 3</code> | Pauses the program for a specified number of seconds. In this case 3 seconds. |
| <code>Send("READ TEMPERATURE 1")</code> | Reads one measurement from the first temperature sensor. |
| <code>Get(T)</code> | Stores the temperature measurement into the variable named T. *Note a get command must immediately follow a read command. The value stored will contain the measurement from the immediately preceding READ command." |
| <code>Output(3,1,"TEMPERATURE= ")</code> <code>Output(3,14,T)</code> | When variable T has a value of 26, "TEMPERATURE= 26" is displayed on line 3 of the calculator. |
| <code>For(N,1,10)</code> <code>Output(3,1,N)</code> <code>End</code> | 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. Note: The step is 1 if the optional step value variable is not used. |
| <code>0→K</code> <code>While K≠45</code> <code>Send("READ TEMPERATURE 1")</code> <code>Get(T)</code> <code>Output(3,1,"TEMPERATURE (°C)= ")</code> <code>Output(3,18,T)</code> <code>getKey→K</code> <code>Wait 1</code> | The commands inside the While structure are looped until the clear key is pressed. The loop continues while logical expression $K \neq 45$, is true. The variable K is initially assigned a value of zero so the While loop will execute at least once. The getKey function monitors the keypad and returns a two-digit value of the row and column of the last key pressed. The clear key is located at row 4 and column 5. |

| | |
|---|--|
| <p>End</p> | |
| <p>If T>25 Then Send("SET COLOR 255 128 0") End</p> | <p>Checks to determine if the value of variable t is greater than 25. If the statement is true then the block of commands in the If..End block are executed. Otherwise the block is skipped. In this case, when the temperature is greater than 26 the calculator will send a command to the TI-Innovator to set the COLOR to be yellow.</p> |
| <p>If T>25 and M<100 Then Send("SET COLOR 255 0 0") End</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> |
| <p>If M<100 Then Output(5,1,"MAGNET IS PRESENT ") Else Output(5,1,"MAGNET IS NOT PRESENT") End</p> | <p>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 100 the commands after Then are executed. In this case the output "MAGNET IS NOT PRESENT" is displayed on row 5. If the value of M is greater than or equal to 100 the commands after Else are executed. In this case the output "MAGNET IS PRESENT " is displayed on row 5.</p> |

Calculator Notes:

- Press the [prgm] key to see a menu option for creating a new program.
- Press the [prgm] key to see a menu option for selecting programs to edit.
- While in the Program Editor, press [prgm] key to see menu items for programs.
- While in the Program Editor, press [alpha] [f5] to see editing options.
- While in Program Editor, use 2nd Left Arrow and 2nd Right Arrow to jump your cursor to the beginning and end of the line.
- Press [2nd] [quit] to leave the Program Editor and return to the home screen.
- While on the home screen, Press the [prgm] key and select the execution menu option for selecting a program names to paste to the home screen.
- While on the home screen, Press [enter] to run a program named on the home screen.
- The Home Screen "remembers" the last command entered. Press Enter after a program has run to run the program again.
- To stop ("break") a running program press and hold the ON key until you receive a dialogue box.

Sensor and actuator Hub connections

