Overview:

In the Pet Car Alarm project, students build and program a simple feedback and control loop. Feedback and control loops are central to many industrial systems and consumer products. The system uses three input modules; Two temperature sensors and a Hall effect magnetic field sensor. The system has four outputs that include two white LED lights (headlights), 1 continuous servo motor (controls the window) and also makes use of the TI-Innovator Hub's built-in speaker (horn honking). The continuous servo requires more power than the calculator can provide. Therefore, an external USB battery is needed. A program is written to read all three input parameters and logically compare them with critical set-point values to determine when to turn on the alarm. The project is presented in a series of small challenges that build the knowledge and skills required for the final open-ended challenge. The final challenge also relies on the students' understanding of important biology and Earth science topics that are relevant to optimizing the system the students ultimately design and refine.

Note: This project assumes the students have a working knowledge of the programming concepts in the Digital Mood Ring project. Please refer to that project as a review of concepts. The materials for the Digital Mood Ring project can be found here - http://bit.ly/2GBIsUN

Possible NGSS topics to explore with students:

Disciplinary Core Ideas:

- HS-LS1-3 – Mechanisms of homeostasis

- MS-ETS1-1 to 1.4 – Engineering design

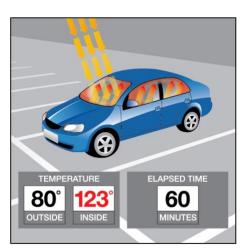- MS-PS3-3 – Design that maximizes thermal energy transfer
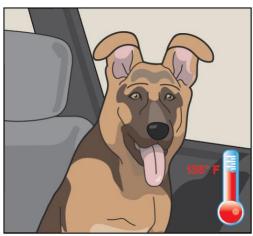
Science & Engineering Practices:

- Asking questions & defining problems

- Evaluate competing designs

- Constructing explanations & designing solutions

Crosscutting Concepts:

- Systems & System Models

- Cause & Effect

- Energy & Matter

Background:

Estimated Vehicle Interior Air Temperature v. Elapsed Time						
Elapsed Time	**Outside Air Temperature (F)**					
	70	**75**	**80**	**85**	**90**	**95**
0 minutes	70	75	80	85	90	95
10 minutes	89	94	99	104	109	114
20 minutes	99	104	109	114	119	124
30 minutes	104	109	114	119	124	129
40 minutes	108	113	118	123	128	133
50 minutes	111	116	121	126	131	136
60 minutes	113	118	123	128	133	138
> 1 hour	115	120	125	130	135	140

Light to Heat

Each summer, there are horrifying stories of children and pets left in hot cars. Ultimately, most end up having a heat stroke and in many cases pass away. The inside of a car heats up so much faster than the outside due to the greenhouse effect. Rays of sunlight stream into the vehicle through the windows and strike, the light strikes the surfaces of the interior of the car. That visible light is absorbed and reradiated as infrared light. Infrared radiation has a larger wavelength than visible light. The infrared radiation is unable to escape back through the windows. The trapped radiation causes the temperature inside the car to rise faster than the outside temperature.

Stayin' Cool!

Mammals such as humans, dogs, and cats all have ways to regulate temperature, thermoregulation, to maintain homeostasis. Humans sweat to increase the removal of heat from the body through evaporation. Dogs usually pant to remove heat although they also have a small number of sweat glands in the pads of their paws. Cats will sprawl out on surfaces that are relatively cool to help remove body heat. They will lick their paws and rub the saliva on warmer parts of their bodies to increase evaporative cooling which is a similar mechanism as sweating in humans. When thermal regulation mechanisms are unable to maintain homeostasis, that mammal will go into heat distress.

The problem with too much heat

Heat distress leads to brain impairment, dehydration, heart failure, cell swelling, and protein denaturation… and potentially death.
Conduct research into why mammals such as humans, dogs, and cats are unable to endure hot environments like closed cars in the summer for long periods. Then design a solution using technology to protect car occupants from heat distress. Refine your design until you have a working prototype. With your teacher's permission, compare your prototype with those of your classmates to determine which team has the "best" system.

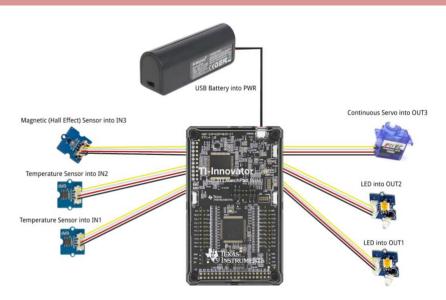| Command	Example	Behavior
CONNECT <type> <number> TO <port>	Send("CONNECT TEMPERATURE 1 TO IN1")	Associates the first TEMPERATURE object with a temperature sensor plugged into port IN1 on the Hub.
SET <type><number> TO <value>	Send("SET LED 1 ON")	Sets LED 1 on . See Hub Settings menu for ON, OFF, BLINK, etc.
SET <type><number> TO <value> [<blink rate>] [<duration in seconds>]	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.
SET <type> <number> TO <value>	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.
Wait <number>	Wait 3	Pauses the program for a specified number of seconds. In this case 3 seconds.
READ <type> <number>	Send("READ TEMPERATURE 1")	Reads one measurement from the first temperature sensor.
Get <variable>	Get(T)	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."
Output(<row 1-10>, <column 1-26>,<expression>	Output(3,1,"TEMPERATURE= ")	When variable T has a value of 26, "TEMPERATURE= 26" is displayed on line 3 of

<variable name> or <string>) **Note:** It is recommended to use the ClrHome command before using Output commands	`Output(3,14,T)`	the calculator.

For <counter variable>,<start value>,<end value>,[<step
value>]
<statements>
End
	`For(N,1,10)`
	`Output(3,1,N)`
	`End`
		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.

While <Boolean expression>
<statements>
End
	`0→K`
	`While K≠45`
	`Send("READ TEMPERATURE 1")`
	`Get(T)`
	`Output(3,1,"TEMPERATURE(°C)= ")`
	`Output(3,18,T)`
	`getKey→K`
	`Wait 1`
	`End`
		The commands inside the While structure are looped until the clear key is pressed. The loop continues while logical expression K ≠ 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.

If <Boolean expression>
Then
<statements>
End
	`If T>25`
	`Then`
	`Send("SET COLOR 255 128 0")`
	`End`
		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.

<Boolean expression> and <Boolean expression>	`If T>25 and M<200`
	`Then`
	`Send("SET COLOR 255 0 0")`
	`End`
		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.

If <Boolean expression>
Then
<statements 1>
Else
<statements 2>
End
	`If M<200`
	`Then`
	`Output(5,1,"MAGNET IS PRESENT    ")`
	`Else`
	`Output(5,1,"MAGNET IS NOT PRESENT")`
	`End`
		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 NOT 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 PRESENT " is displayed on row 5.

See TI-Innovator Technology eGuide for more background https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html

Setup Project:

Supplies:

- Calculator
- Unit to Unit Cable
- TI-Innovator Hub
- Grove Temperature sensor x2 (inside and outside measurement)
- Grove Hall effect magnetic proximity sensor
- Grove White LED Light x2
- Grove Continuous Servo motor
- Grove Cable x5
- External USB Battery w/ Micro USB Cable
- Fashion Doll Car, shoebox, or another object to model a car
- Small toy pet
- Small magnet (ceramic, ferrite or ceramic-ferrite) for pet collar
- Small piece of clear plastic to model car window
- Cellophane tape to attach magnet to pet and window to motor
- Cling wrap (e.g. Saran) to enclose model car
- Safety scissors

Student Activity:	Teacher Notes:

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")

Teacher Guidance during Challenge 1:
- Use a For..End loop to alternate between two different sound frequencies one octave apart. Loops are used to repeat a set of command. A For loop repeats a specified number of times. In TI-Basic the programmer defines a For loop with four inputs: a counter variable, a beginning value for the counter variable, an ending value for the counter variable and an optional step value variable.
- For and End are found on the Program Control Menu.

- Use the Wait command to determine how long each tone plays

- Example program:

```
PROGRAM: PC1
For(N,1,10)
Send("SET SOUND 440")
Wait 1
Send("SET SOUND 880")
Wait 1
End
```

- Try adding a third tone in your program
- Try to find the limits of the built-in speaker by changing the frequencies

Challenge 2a: Use a For..End loop to blink two external LED's.

Teacher Guidance during Challenge 2:

- The Hub Connect commands provide the hub with information about the type of device plugged into external ports and how to communicate with that device.
- Connect LED's to OUT 1 and OUT 2 on the Hub
- Use the Send "CONNECT LED # TO OUT #" command.
- Use a For..End loop to make the LED's turn on and off.
- Example program:

```
PROGRAM: PC2A
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
```

```
End
```

Challenge 2b: Try Challenge 2A once more using the "BLINK" and "TIME" options. This time you can do it without a For loop.

- Example program:

```
PROGRAM: PC2B
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
```

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).

Teacher Guidance during Challenge 3:

- Use a For..End loop to cause a continuous servo motor to rotate clockwise and then counterclockwise.
- Use Send "CONNECT SERVO.CONTINUOUS # TO OUT 3"
- The SERVO.CONTINUOUS motor must be connected to the OUT 3 port because 5 volts are necessary to power the servo motor.
- Use Clockwise (CW) or Counter Clockwise (CCW) to set the turn direction of the continuous servo.
- Power for the continuous servo motor is ranged from 0 to 100; 0 being off and 100 being full power.
- Be sure external battery is attached to the PWR port of the TI-Innovator Hub and turned on just before running program.
- Example program

```
PROGRAM: PC3
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
End
```

- Experiment with changing the power and TIME duration and the associated wait time to see what effect that has on how many turns the servo will make.

Challenge 4: Connect a temperature sensor to the TI-Innovator Hub and display

Teacher Guidance during challenge 4:

- The Hub Connect commands provide the hub with information about the type of device plugged into

the temperature on the calculator.

external ports and how to communicate with that device.

- The Read command instructs the Hub to read a value from a device plugged into a specified port. The Hub places the value in a memory location on the Hub, waiting for the calculator to "get" the value.
- The Get command gets the value from the Hub memory location for values read and stores the value to a specified variable.
- While loops are useful when you would like a set of commands to be executed when a certain condition is true.
    - The While loop test checks to see if the variable named K is not equal to 45. If K is not equal to 45 then the commands in the loop are run. If K is equal to 45 then the program exits the While loop and moves to the next command after the End statement. In the example, the variable named K is initially set to be 0. This assures that the While loop will run at least once.
    - In the While loop a function called getKey is used to update the value stored in the variable named K. getKey is a TI-84 function that is useful for keyboard controlled While loops. getKey stores a 2-digit code value for the last key that you pressed while a program is running. The first digit is the row on the keyboard of the key, counting from the top. The second digit is the column of the key location on the keyboard. In this example, there is a check to see if the clear key, code value of 45, has been pressed.
- While and End are found on the Program Control Menu.
- Example program:

```
PROGRAM: PC4
Send("CONNECT TEMPERATURE 1 TO IN 1")
ClrHome
Output(8,1,"PRESS CLEAR TO QUIT")
0→K
While K≠45
Send("READ TEMPERATURE 1")
Get(T)
Output(3,1,"TEMPERATURE(°C)= ")
Output(3,18,T)
Wait 0.5
getKey→K
End
```

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.

Teacher Guidance during Challenge 5:

- Some sensors or output devices do not have an associated object that can be found in the input and output menus. For these devices we use ANALOG.IN, DIGITAL.IN, ANALOG.OUT, or DIGITAL.OUT.
- ANALOG.IN 1 is the object for the Hall effect magnetic proximity sensor.
- The Hall effect sensor detects the presence (or absence) of a magnetic field with south polarity. For the Hall sensor you will use ANALOG.IN.
- Sensor values are related to voltage measurements. When the south pole of the magnet is close, readings are usually under 200.
- It is recommended that IN 3 is used to supply enough power to the sensor.
- Example program:

```
PROGRAM: PC5
Send("CONNECT ANALOG.IN 1 TO IN 3")
ClrHome
Output(8,1,"PRESS CLEAR TO QUIT")
0→K
While K≠45
Send("READ ANALOG.IN 1")
Get(M)
Output(3,1,"MAGNETIC SENSOR(BITS)= ")
Output(4,1,"     ")
Output(4,1,M)
If M<200
Then
Output(5,1,"MAGNET IS PRESENT     ")
Else
Output(5,1,"MAGNET IS NOT PRESENT")
End
Wait 0.5
```

```
getKey→K
End
```

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.

Teacher Guidance during Final Challenge:

- This challenge has students create a system that reads data from inputs such as temperature and the Hall effect sensor which in turn controls LED's, sounds through the speaker, and the continuous servo motor.
- Example program:

```
PROGRAM: PCALARM
Send("CONNECT TEMPERATURE 1 TO IN 1")
Send("CONNECT TEMPERATURE 2 TO IN 2")
Send("CONNECT ANALOG.IN 1 TO IN 3")
Send("CONNECT LED 1 TO OUT 1 ")
Send("CONNECT LED 2 TO OUT 2 ")
Send("CONNECT SERVO.CONTINUOUS 1 TO OUT 3")
ClrHome
Output(1,1,"PET CAR ALARM NOW ON")
Output(8,1,"PRESS CLEAR TO QUIT")
1→W
0→K
While K≠45
Send("READ TEMPERATURE 1")
Get(I)
Send("READ TEMPERATURE 2")
Get(O)
Send("READ ANALOG.IN 1")
Get(M)
If M<200
Then
Output(3,1,"PET IN CAR     ")
Else
Output(3,1,"PET NOT IN CAR")
End
If W=1
```

```
Then
Output(4,1,"WINDOW IS CLOSED")
Else
Output(4,1,"WINDOW IS OPEN  ")
End
Output(5,1,"TEMPERATURE INSIDE(°C)= "
Output(5,25,I)
Output(6,1,"TEMPERATURE OUTSIDE(°C)= "
Output(6,26,O)
If I>25 and M<200
Then
Output(2,1,"PET IS IN HEAT DISTRESS")
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 W=1
Then
Send("SET SERVO.CONTINUOUS 1 CW 30 .25")
Wait .25
0→W
End
Else
Output(2,1,"NO PROBLEMS DETECTED   ")
Send("SET LED 1 OFF")
Send("SET LED 2 OFF")
Send("SET SOUND OFF")
If W=0
Then
Send "SET SERVO.CONTINUOUS 1 CCW 30 .25"
```

```
Wait .25
1→W
End
End
getKey→K
End

Send("SET LED 1 OFF")
Send("SET LED 2 OFF")
Send("SET SOUND OFF")
Send("SET SERVO.CONTINUOUS 1 0")
ClrHome
Output(1,1,"PET CAR ALARM IS OFF")
```