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 - Digital Mood Ring Project

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

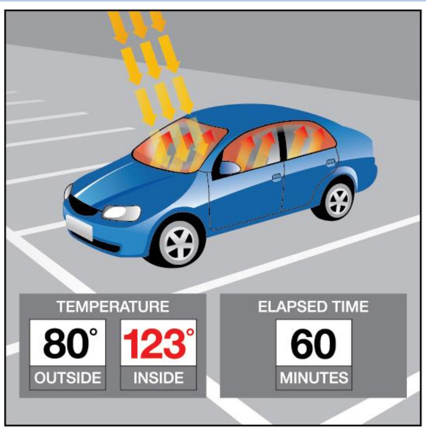
Science & Engineering Practices:

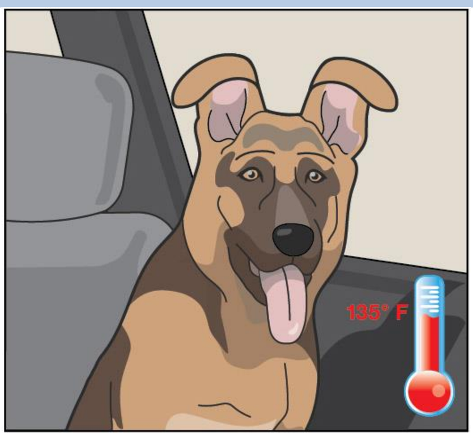
- Asking questions & defining problems
- Evaluate competing designs
- Constructing explanations & designing solutions

Crosscutting Concepts:

- Systems & System Models
- Cause & Effect
- Energy & Matter

Background:





Estimated Vehicle Interior Air Temperature v. Elapsed Time

Inside Air Temperature (F)

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.

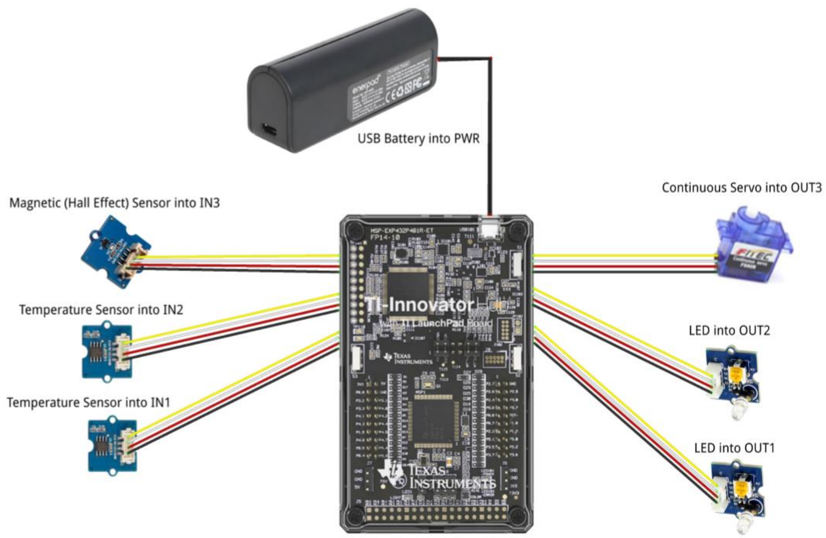
Command	Example	Behavior
CONNECT <type> <number> TO <port>	`Send("CONNECT TEMPERATURE 1 TO IN1")`	Associates the first TEMPERATURE object with a temperature module plugged into port IN1 on the Hub.
SET <type><number> TO <value>.	`Send("SET LED 1 TO ON")`	Turns on LED 1. Other parameters may include BLINK <rate> TIIME<duration> "SET LED 1 ON BLINK 3 TIME 20"
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 a measurement from the first temperature object.
Get(<variable>)	**Get(*t*)**	Stores the temperature measurement into a variable named *t*. The value stored will contain the measurement from the immediately preceding READ command. Note: a Get command must immediately follow a READ command.
Output(<line #> ,<column#>, <"text">)	**Output(3,1,"Temperature = ")** **Output(3,15, T)**	When variable *t* has a value of 26, the following line is displayed on the calculator: Temperature = 26 (temp readings are in °C by default)
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.
While <Boolean expression> <statements> End	**1 ⟶ K** **While *K* ≠ 45** **READ("TEMPERATURE 1")** **Get(*t*)** **Output(3,1,"Temperature = ")** **Output(3,15, T)** **getKey ⟶ K** **Wait 1** **End**	The statements in the While loop are executed until the clear key is pressed. The While loop continues as long as the Boolean expression evaluates to "true". The variable *K* is set to an initial value of 1 using the assign function ⟶. getKey is a function that returns a string with the integer value of the last key pressed while a programming is running. In this program the value of getKey is stored to the variable *K*.
<Boolean expression> and <Boolean expression>	**If *t*>25 and *m*<100** **Then** **Send "SET COLOR 0 255 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<100** **Then** **Output(5,1,"Magnet present")** **Else** **Output(5,1,"Magnet 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 100 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 100 the commands after Else are executed. In this case the output “Magnet is not 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:



Materials:

- Calculator
- Unit to Unit Cable
- TI-Innovator Hub
- Grove Cable x5
- Grove Temperature sensor x2
- Fashion Doll Car, shoebox, or another object to model a car
- Grove Hall effect magnetic proximity sensor
- Grove White LED Light x2
- Grove Continuous Servo motor
- External USB Battery w/ Cable

Student Activity:

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”)

Extension: Write a program that will play

Teacher Notes:

Teacher Guidance during Challenge 1:

The background section explains different tones have different frequencies. It explains each note’s frequency can be found by multiplying the pervious note by the twelfth root of 2. It tells students middle C on a piano has a frequency of 261.64 Hz.

- Students will Use the Send(“Set Sound”) command to determine:
  1) the lowest frequency you can set and still hear it___________
     (Answers will vary between 5 and 50)
  2) the highest frequency you can set___________
     (Answers will vary around 7937)

through an entire octave. Can you use a loop with math so you don't have to use 12 different "SET SOUND" commands?

- 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…End is found on the Program Control Menu
- Use the Wait command to determine how long each tone plays
- Example program for Challenge 1:

```
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*
- Example program for **Extension**:
  This question is not essential to the Pet Car Alarm project. It is however, a nice music extension that uses the mathematical relationships described in the background knowledge. Students will need to use the eval() function inside the Send function to evaluate their variable. There are numerous possible solutions to the problem. Two different possible solutions are provided below.

```
PROGRAM: OCTAVE
261.64 → F
For(N,1,12)
 Send("SET SOUND eval(F) TIME 1")
 Wait 1
 F*(2^(1/12)) → F
End
```

```
PROGRAM: OCTAVE
261.64→  F
For(n,0,11)
 Send "SET SOUND eval(F*(1.0595)^n)
End
```

- *Note: A1 students may be more comfortable with the first solution, but A2 students may be more comfortable with the second solutions that use exponents.*

Challenge 2:

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

Teacher Guidance during Challenge 2:

Background section discusses the use of pre-programmed lights:.

Does your phone light up when someone texts? Withouth turn signals how would other motorists know to slow when a car in front of them plans to turn? Some lights such as the one in the refridgerator turn on and off with actions such as opening and closing the door. Other lights, such as traffic lights, cycle through a set pattern. Some lights are programed with an action/consequence plan meaning they don't change their state until something occurs. Others are pre-set to run in a pattern. Have students brainstorm with a partner various lights they encounter throughout the day.

1) What are three lights you encounter in a day that change based on an action?
Sample Answers: Headlights or interior lights in a car might automatically turn on when the car is started or the door is opened. Some night lights turn themselves on when the room is dark. Some lights are motion activated and turn themselves on when motion is detected or off after a given amount of time with no motion detected. Cell phones might use light to indicate an incoming call or text message.

2) What are three lights you encounter in a day that have a preset pattern?
Sample Answers: Cross walk signals can light up without an action they change as traffic lights change. Digital clocks use lights in a pattern to display time. Turning signals on a vehicle have a preset pattern when they are turned on.

Challenges 2a/b:

- 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 command to make the LED's turn off and on

- 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
End
```

Extension 1: Write a program that lets the user press 1 to turn on the external LED's and 0 to turn them off. Your IF THEN statements should be inside a while loop that continues to execute until the user presses esc.

- Both Extensions are optional.

```
PROGRAM: PC2EXT1
Send("CONNECT LED 1 TO OUT 1 ")
Send("CONNECT LED 2 TO OUT 2 ")
1 → K
While K ≠ 45
  If K = 92
  Then
    Send("SET LED 1 ON")
    Send("SET LED 2 ON")
  End
  If K = 102
```

```
  Then
    Send("SET LED 1 OFF")
    Send("SET LED 2 OFF")
  End
  getKey → K
End
```

Extension 2: Have you ever heard of Morse Code? Morse Code uses a combination of dots and dashes to represent letters and numbers. An S in Morse Code is · · · (dot dot dot) while an O is - - - (dash dash dash). Sending an SOS is the international sign of distress. Write a program that sends out an SOS using three loops, one for each letter.

```
PROGRAM: PC2EXT2
Send("CONNECT LED 1 TO OUT 1 ")
Send("CONNECT LED 2 TO OUT 2 ")
For(N,1,3)
  Send("SET LED 1 ON")
  Send("SET LED 2 ON")
  Wait 0.5
  Send("SET LED 1 OFF")
  Send("SET LED 2 OFF")
  Wait 0.5
End
For(N,1,3)
  Send("SET LED 1 ON")
  Send("SET LED 2 ON")
  Wait 1.5
  Send("SET LED 1 OFF")
  Send("SET LED 2 OFF")
  Wait 0.5
End
For(N,1,3)
  Send("SET LED 1 ON")
  Send("SET LED 2 ON")
  Wait 0.5
  Send("SET LED 1 OFF")
  Send("SET LED 2 OFF")
```

```
  Wait 0.5
End
```

Challenge 3:

Challenge 3a: 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:

Background section:

Have you ever wondered how heating and cooling systems work? Have you ever changed the thermostat setting in a room or a car? Some heating and cooling systems require the use to turn knobs to turn them on and off. The user must actively switch on and off the heating and cooling system. Other thermostats are temperature controlled. When the temperature reaches the predetermined level, the heating or cooling mechanism is activated, and the user doesn't have to manually turn them on and off. In this activity, students will program the calculator to use a sensor to measure the temperature and control a servo motor synthesizing turning on and off a fan.

- 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: PC3A
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.
```

Challenge 3b:
Connect a temperature sensor to the TI-Innovator™ Hub and display the temperature on the calculator.

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.

- 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.
- 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, the clear key. 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 a 1. 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-Nspire function that is useful for keyboard controlled While loops. getKey stores the value of the last key that you pressed while a program is running. The key values are integers. For example, the While loop test checks for 45 not clear.
- Note that multiple arguments can be used to build a message with the Output command.
- Example program:

```
PROGRAM: PC3B
Send("CONNECT TEMPERATURE 1 TO IN 1")
1 →  K
While K ≠ 45
  Send("READ TEMPERATURE 1")
  Get(t)
  Output(3,1,”TEMPERATURE =”)
  Output(3,15,T)
  Wait 0.5
  getKey →  K
End
```

Challenge 3c:
Create a program that triggers the turning on/off of an air conditioning system.
If the temperature is too warm, turn the servo motor on for 2 seconds.

```
PROGRAM: PC3C
Send("CONNECT SERVO.CONTINUOUS 1 TO  OUT 3 ")
Send("CONNECT TEMPERATURE 1 TO  IN 1 ")
1 → K
While K ≠ 45
  Send("READ TEMPERATURE 1")
  Get(T)
  Output(3,1,”TEMPERATURE =”)
  Output(3,15,T)
  If T>27
  Then
      Send("SET SERVO.CONTINUOUS 1 CW 20 TIME 1")
      Wait 2
  End
  Wait 0.5
  getKey → K
End
```

Challenge 4: 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 4:
The background section discusses warning systems that students may encounter on a daily basis:

Have you ever sat in the passenger seat of a car and heard the warning system indicating you needed to buckle your seatbelt? Why does this alarm only go off if there is someone in the seat? How does the warning system know when there is a person in the seat? Are there times this system could mistakenly think a person is in the seat and sound the alarm, but really there isn’t someone in the seat?

- *Can you think of another warning system you’ve encountered? What type of information does it use to determine if the system should sound an alarm?*
  (Answers will vary. Some possible answers could be a door ajar warning light if the car door isn’t closed all the way. Smoke dectors that go off when then sense smoke.)

The Hall Effect sensor students will use in this activity has values that are related to voltage measurements. The readings are usually under 100 when the south pole of the magnet is close to the sensor.

- *Look at the side of your TI-Innovator:*
  - *What are the voluts for IN 3?*
  - *What are the volts for IN 1 and IN 2?*
  - *The temperature sensors can go into IN 1 and IN 2 while the Hall Effect sensor is recommended for IN 3, what does this tell you about the requirements of the temperature sensor compared to the Hall Effect sensor? (Answer: The Hall Effect sensor requires more volts.*
- **Put the Hall Effect sensor in IN 3 to supply enough power for this type of sensor.**

- 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.
- 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 100.
- It is recommended that IN3 is used to supply enough power to the sensor.
- Note: The © symbol denotes a comment. Comments are not executed when the program is run. Use the Program/Actions menu to add a comment.

Example program:

```
PROGRAM: PC4
Send("CONNECT ANALOG.IN 1 TO IN 3”)
1 →  K
While K ≠ 45
  Send(“READ ANALOG.IN 1”)
  Get(M)
  Output(3,1,”Magnet: ”)
  Output(3,9,T)
  If M<100
  Then
    Output(5,1,”Magnet Present    ”)
  Else
```

```
    Output(5,1,"Magnet NOT Present")
  End
  Wait 0.5
  getKey → K
End
```

Challenge 5 (Final): 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 is the culminating challenge, and has students create a system that reads data from inputs such as temperature and the hall sensor which in turn controls LED's, sounds through the speaker, and the continuous servo motor.
- Notice that While..End is used often with sensors (inputs) and If..End is used with outputs.
- Example program:

```
PROGRAM: PC5
Output(1,1,"Pet Car Alarm ON ")
1 → K
1 → W
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")
While K ≠ 45
  Send("READ TEMPERATURE 1")
  Get(T)
  Send("READ TEMPERATURE 2")
  Get(O)
  Send("READ ANALOG.IN 1")
  Get(M)
  If M<100
```

```
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,"Temp in car       ")
Output(5,13,T)
Output(6,1,"Temp outside      ")
Output(6,14,O)
If T>25 and M<100
Then
  Output(2,1 "Warning! 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 warning        ")
```

```
    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")
Output(1,1,"Pet Car Alarm OFF")
```