Project: Making Music with Code

PROJECTS WITH THE TI-INNOVATOR™ SYSTEM - TI-84 PLUS CE

Overview:

Students will write a TI-BASIC program on their calculator to create sounds using the built-in speaker on the TI-Innovator Hub. They will also experiment with changing the length of time in which each sound plays. Once they have some experience, they will write a program to play a song.

Goals:

Students will:

- learn to create, edit and run programs from the editor.
- write a TI BASIC program that controls the built-in speaker on the TI-Innovator Hub
- write a program that controls how long a sound will play.

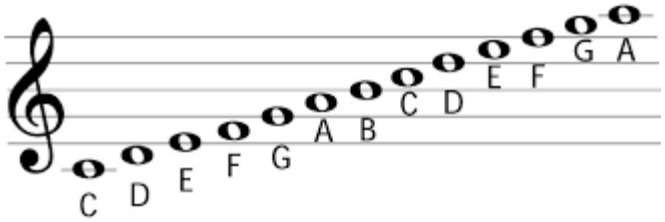
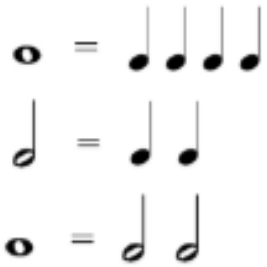
Note – A good foundation before beginning this project can be found at https://education.ti.com/en/activities/ti-codes/84/10-minutes-innovator
Go to Unit 1: Getting Started with TI-Innovator Hub and then "Skill Builder 3 – Input command and SOUND". Also, Unit 2: For Loops with TI-Innovator Hub and then "Skill Builder 3 Loop through the musical notes" has related concepts that may be helpful.

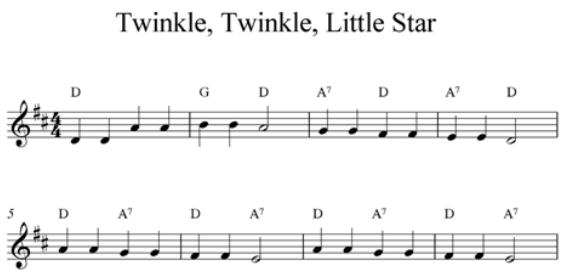
Background:

Command	Example	Behavior
SET SOUND <frequency> TIME <duration>	Send("SET SOUND 440 TIME 4")	Plays a 440 hertz tone on the speaker for 4 seconds
Output(<row 1-10>, <column1-26>, <variable name> <expression>or <string in quotes>) **Note:** It is recommended to use the ClrHome command before using Output commands	`Output(3,1,"FREQUENCY(HZ)= ")` `Output(3,17,F)`	When variable F has a value of 440, the following line is displayed on the calculator: FREQUENCY(HZ) = 440
For(<counter variable>,<start value>,<end value>,[<step value>] <statements>) End	For(N,1,10) End	Runs the statements within the For/End loop 10 times
eval(<expression> or <variable name>)	440→N Send("SET SOUND eval(N)")	When eval is used within a Hub command, the variable value is inserted into the Hub command. In the example, the value 440 is stored to variable N. The next command is evaluated as Send("SET SOUND 440")and a tone is played on the Hub.
Wait <duration>	Wait 2	Causes program execution to halt and wait for two seconds.
"COMMENT ABOUT THE PROGRAM"	"NEXT COMMAND PLAYS C4"	Quotes without a preceding command in a TI-84 Plus CE program can be used as a comment. A comment is meant to leave notes within the program for easier navigation later or to let the user know what the following line of code is used for. Comments are ignored when the program is run.

©2018 Texas Instruments Incorporated

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

Music Concepts	Explanation
Musical Scale 	Music is arranged with symbols on a "scale". We use a scale since the notes are divided by "octaves". An octave represents eight steps between the equivalent note that is at either double or half the frequency. For example, A4 is 440 hertz (Hz), while A5 is 880 Hz (twice the value of A4), while A3 is 220 Hz (half of A4).
Beats per minute/Tempo	Have you ever noticed how some songs are slow while others are fast? Every song has its own "tempo" or speed. The speed is determined by the number of "beats per minute". A slow song may have 50 beats per minute while a faster song may have 100 beats per minute.
Quarter, half, and whole notes 	Notes can be shorter or longer depending on the song. One way for a musician to write a song with notes of different durations is to use whole notes, half notes, quarter notes, eighth notes, and sixteenth notes. One whole note equals four quarter notes or two half notes. One half note is equal to two quarter notes and so on. For a song that has 4-4 measures, it means that each measure of the song has 4 beats. A whole note would be equal to 4 beats (hence, one measure). How many half notes could fit into a 4-4 measure? If you said two, you are correct! Since each half note is equal to two beats and there are four total beats per measure, two half notes is equal to four beats. So now you can see how a song with 100 beats per minute will have faster whole notes than a song with 50 beats per minute! Not all whole notes equal the same time duration!

Application of notes and beats per minute 	Twinkle, Twinkle, Little Star is a song that has multiple notes and uses 4-4 time. Try to code this song with your calculator.
Note - Another useful resource for understanding musical concepts, frequencies and notes can be found http://www.sengpielaudio.com/calculator-notenames.htm	
♩ = 100 Free violin sheet music at http://www.capotastomusic.com	

Setup Project:	**Supplies:**
Students may work in groups of two or three.	• Calculator • Unit to Unit Cable • Hub

The Student Activity	**The Teacher Activity**
Sit in small groups with your calculator and supplies for this activity. Practice the guidance modeled by your teacher.	Review and introduce the calculator and Hub commands needed for this activity. • Start a new document • Insert a program • Attach Hub

Challenge 1: Create a program called "**C1**". Set the speaker on the Hub to play a tone of 440 Hz. Try several different tones. What is the lowest tone you can hear? What is the highest?

Guidance during challenge 1:

- Use the Output command (Program I/O menu). Output requires three inputs (arguments). The first arguments define the row (1-10) and character (1-26) to begin the output from. The third argument is the text string, variable or expression to display.
- Example program

```
PROGRAM:C1
ClrHome
Send("SET SOUND 440 TIME 2")
Output(3,1,"THIS IS 440HZ FOR 2 SECONDS")
```

Challenge 2: Create a program called "**C2**". Make a door bell using any key press on the calculator to ring the chime.

Guidance during challenge 2:

- By the way, most common door bells sound two tones that are a perfect 4th apart. An example is D#5 and G5.
- Example program

```
PROGRAM:C2
"G5"
Send("SET SOUND 784")
Wait 1
"D#5"
Send("SET SOUND 622")
```

Challenge 3: Write a program called "**C3**" that plays the Major C scale from C4 to C5. Each note should be played for a variable length of time using a variable named D (for duration).

Guidance during challenge 3:

- Example program

```
PROGRAM:C3
0.1→D
"C4"
Send("SET SOUND 262 TIME eval(D)")
Wait D
```

©2018 Texas Instruments Incorporated

```
"D4"
Send("SET SOUND 294 TIME eval(D)")
Wait D
"E4"
Send("SET SOUND 330 TIME eval(D)")
Wait D
"F4"
Send("SET SOUND 349 TIME eval(D)")
Wait D
"G4"
Send("SET SOUND 392 TIME eval(D)")
Wait D
"A4"
Send("SET SOUND 440 TIME eval(D)")
Wait D
"B4"
Send("SET SOUND 494 TIME eval(D)")
Wait D
"C5"
Send("SET SOUND 523 TIME eval(D)")
Wait D
```

Challenge 4: Create a program called "**C4**". Make a metronome that: plays 10 beats with a tone of middle C4 with a duration of a ¼ note at 100 Beats per minute. To do this you will need to determine the how long a quarter note is (in seconds) based on the information

Guidance during challenge 4:

- Loops are used to repeat a set of commands. A "For loop" repeats a specified number of times. In TI-Basic the programmer defines a For loop with four arguments: a counter variable, a beginning value for the counter variable, an ending value for the counter variable and an optional step value. The step value is 1 if a value is not entered.
- Students will need to determine the length of time per beat at 100 beats per minute for a quarter note (which is one beat). They can do this by dividing 60 seconds/minute by 100 beats per minute, which

above.

A rest of .1 seconds should be used between beats. Try a double time tempo of 200 BPM. Try using a 1/2 or whole note.

*Use the file “88 Piano Keys” utility found in 88 Piano Keys.tns file or create tables to convert notes to duration and frequency.

equals 0.6 seconds/beat.

- Example program

```
PROGRAM:C4
"1/4 NOTE AT 100BPM"
0.6→D
For(N,1,10)
 Send("SET SOUND 262 TIME eval(D)")
 Wait D
 Wait 0.1
End
```

Challenge 5: Create a program called “**C5**”. Use the sheet music for Twinkle to play the first two bars of the song on your Hub.

*Use the file “88 Piano Keys” utility found in 88 Piano Keys.tns file or create tables to convert notes to duration and frequency.

Guidance during challenge 5:

- Example program

```
PROGRAM:C5
Send("SET SOUND 261 TIME .6")
Wait 0.6
Wait 0.1
Send("SET SOUND 261 TIME .6")
Wait 0.6
Wait 0.1
Send("SET SOUND 392 TIME .6")
Wait 0.6
Wait 0.1
Send("SET SOUND 392 TIME .6")
Wait 0.6
Wait 0.1
Send("SET SOUND 440 TIME .6")
Wait 0.6
Wait 0.1
Send("SET SOUND 440 TIME .6")
Wait 0.6
```

```
Wait 0.1
Send("SET SOUND 392 TIME 1.2")
Wait 1.2
Wait 0.1
```

Challenge 6: Create a program called "**C6**". Use two lists with frequencies and durations and a loop to play the same Twinkle song.

Guidance during challenge 6:

- Lists variables are a set of values. On the TI-84 Plus CE the values are separated by commas and enclosed by braces. The calculator has 6 pre-defined list variables named L_1-L_6. The dim(<listname>) function found on the List Ops (Operations) menu returns the length (dimension) of a list. TI-84 Plus CE list values can be accessed by the listname followed the by the position in the list within parentheses starting at 1. For example, $L_1(1)$ would return the first value of the list variable L_1.
- Example program

```
PROGRAM:C6
"MAKE LISTS L1 NOTES L2 DURATIONS"
{261,261,392,392,440,440,392}→L1
{0.6,0.6,0.6,0.6,0.6,0.6,1.2}→L2
For(N,1,dim(L1))
 Send("SET SOUND eval(L1(N)) TIME eval(L2(N)")
 Wait L2(N)
 Wait 0.1
End
```

Final Challenge: Find the sheet music for a simple song and play it on the Hub.

Guidance during Final Challenge:

- Have the students use the skills that they have built from the previous challenges.
- Develop a program that can play a well-known song.
- Example Program

```
PROGRAM:FINAL
"MAKE LISTS L1 NOTES L2 DURATIONS"
{--,--,--,--,--,--,--} →L1
{--,--,--,--,--,--,--} →L2
```

```
For(N,1,dim(L₁))
 Send("SET SOUND eval(L₁(N)) TIME eval(L₂(N)")
 Wait L₂(N)
 Wait 0.1
End
```